

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

(підпис) О.А.Павлов
(ініціали, прізвище)

“ ” 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 «Програмна інженерія»

на тему Онлайн – платформа для надання юридичних послуг

Виконав:
студент IV курсу,
групи

ІП-51 Субцельний Олександр Володимирович
(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Муха І. П.
посада, науковий ступінь, вчене звання, прізвище, ініціали

(підпис)

**Консультант
з графічної
документації**

доц., к.т.н. Лишук К.І.
посада, науковий ступінь, вчене звання, прізвище, ініціали

(підпис)

Рецензент:

доц., к.т.н., доц. Ткач М. М.

посада, науковий ступінь, вчене звання, прізвище, ініціали

(підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – **6.050103**
«Програмна інженерія» (Програмне забезпечення систем)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

О.А. Павлов
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Субцельному Олександру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема проекту «Онлайн – платформа для надання
юридичних послуг»

керівник проекту Муха Ірина Павлівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. №1181-с

2. Термін подання студентом проекту «03» червня 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

*1) Аналіз вимог до програмного забезпечення: основні визначення та терміни,
опис предметного середовища, огляд існуючих технічних рішень та відомих*

програмних продуктів, розробка функціональних та нефункціональних вимог
2) Моделювання та конструювання програмного забезпечення: моделювання та
аналіз програмного забезпечення, засоби розробки, технічні рішення, архітектура
програмного забезпечення

3) Аналіз якості та тестування програмного забезпечення: опис процесів
тестування

4) Впровадження та супровід програмного забезпечення.

5. Перелік графічного матеріалу

1) *Схема структурна розгортання*

2) *Схема структурна компонентів програмного забезпечення*

3) *Схема бази даних*

4) *Схема структурна класів програмного забезпечення*

5) *Креслення вигляду екранних форм*

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «12» березня 2018 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>01.04.2019</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>10.04.2019</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>15.04.2019</i>	
4.	<i>Аналіз вимог до програмного забезпечення</i>	<i>18.04.2019</i>	
5.	<i>Алгоритмізація задачі</i>	<i>21.04.2019</i>	
6.	<i>Моделювання програмного забезпечення</i>	<i>23.04.2019</i>	
7.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>27.04.2019</i>	
8.	<i>Розробка архітектури програмного забезпечення</i>	<i>02.05.2019</i>	
9.	<i>Розробка програмного забезпечення</i>	<i>18.05.2019</i>	
10.	<i>Налагодження програми</i>	<i>20.05.2019</i>	
11.	<i>Виконання графічних документів</i>	<i>22.05.2019</i>	
12.	<i>Оформлення пояснювальної записки</i>	<i>24.05.2019</i>	
13.	<i>Подання ДП на попередній захист</i>	<i>28.05.2019</i>	
14.	<i>Подання ДП рецензенту</i>	<i>31.05.2019</i>	
15.	<i>Подання ДП на основний захист</i>	<i>03.06.2019</i>	

Студент _____ Субцельний О. В.
(підпис)

Керівник проекту _____ Муха І. П.
(підпис)

[illegible]

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з чотирьох розділів, містить 89 сторінок, 18 рисунків, 36 таблиць, 8 джерел.

Дипломний проект присвячений створенню онлайн - платформи для надання юридичних послуг в частині розробки веб-сервісу для реалізації взаємодії консультанта з платформою, REST API для мобільних додатків, що реалізують взаємодію клієнта з даною платформою, адміністративної панелі для керування платформою та інтегрування зовнішніх сервісів для підтримки дієздатності платформи.

У розділі з аналізу вимог до програмного забезпечення описано предметне середовище, розроблені функціональні вимоги для кожного з акторів та нефункціональні вимоги для платформи.

У розділі з моделювання та конструювання програмного забезпечення розроблено принципову схему клієнт – серверної моделі, описано і розроблено API для мобільних платформ та побудовано загальну архітектуру платформи.

У розділі з аналізу якості та тестування програмного забезпечення надано опис процесів тестування.

У розділі з впровадження та супроводу програмного забезпечення описаний процес розгортання платформи.

ОНЛАЙН-ПЛАТФОРМА, ВЕБ-СЕРВІС, ЮРИДИЧНІ ПОСЛУГИ, КОНСУЛЬТАЦІЇ, КЛІЄНТСЬКЕ ЗАСТОСУВАННЯ, АДМІНІСТРАТИВНА ПАНЕЛЬ, REST API

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ABSTRACT

The structure and scope of the thesis. The explanatory note of the diploma project consists of 4 chapters, 89 containing pages, 18 drawings, 36 tables, 8 sources.

The diploma is devoted to the creation of an online legal services platform namely the development of a web service for the implementation of the interaction of a platform and the consultant, the REST API for mobile applications implementing the client's interaction with this platform, the administrative service for managing the platform and integrating external services to maintain the capacity of the platform.

In the section on software requirements analysis the subject matter is described, functional requirements are developed for each of the actors, and non-functional requirements for the platform.

In the section on modeling and designing of the software the basic scheme of the client-server model has been developed, the API for mobile platforms has been developed and the general architecture of the platform has been built.

The section on quality analysis and software testing a description of the testing processes is provided.

The section on software implementation and maintenance describes the process of the platform deployment.

ONLINE PLATFORM, WEB SERVICE, LEGAL SERVICES, CONSULTING, CLIENT APPLICATION, ADMINISTRATIVE PANEL, REST API.

Пояснювальна записка до дипломного проекту

на тему: ОНЛАЙН – ПЛАТФОРМА
ДЛЯ НАДАННЯ ЮРИДИЧНИХ ПОСЛУГ

Київ – 2019 року

					КПІ.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП.....	11
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	13
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ.....	14
1.3.1 Аналіз відомих технічних рішень.....	14
1.3.2 Аналіз відомих програмних продуктів.....	14
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	16
1.4.1 Розроблення функціональних вимог.....	18
1.4.2 Розроблення нефункціональних вимог.....	39
1.4.3 Постановка комплексу завдань.....	40
1.5 Висновки по розділу	42
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	43
2.1. МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	43
2.1.1 Частина користувача.....	43
2.1.2 Частина консультанта.....	49
2.1.3 Адміністративна панель.....	57
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	60
2.3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	61
2.4 АНАЛІЗ БЕЗПЕКИ ДАНИХ.....	74
2.5 Висновки по розділу	75
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	76
3.1 АНАЛІЗ ЯКОСТІ ПЗ.....	76
3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....	77
3.2.1 Тестування конвертації та обміну даними.....	78
3.2.2 Тестування зручності інтерфейсу.....	78
3.2.3 Тестування інтерфейсу користувача.....	78
3.2.4 Тестування функціоналу сайту та додатку.....	79

3.2.5	Тестування навантаження.....	79
3.2.6	Тестування безпеки.....	79
3.2.7	Тестування стабільності.....	79
3.3	ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ	80
3.4	ВИСНОВКИ ПО РОЗДІЛУ	85
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	86
4.1	РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	86
4.2	РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	86
4.3	ВИСНОВКИ ПО РОЗДІЛУ	86
	ВИСНОВКИ	88
	ПЕРЕЛІК ПОСИЛАНЬ.....	89

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – (Application Programming Interface, API) — набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено - це набір чітко визначених методів для взаємодії різних компонентів.

JSON – це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною.

Токен – послідовність випадкових символів, яка використовується для підтвердження дії користувача.

JWT - JSON Web Token

					КПІ.ІП-5121.045440.01.81	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Неможливо знати все: ніхто не завчає норми законів напам'ять, складно бути в курсі всіх останніх новин та оновлень, зважаючи на швидкість їх потоку, в той же час займає чимало часу пошук необхідної інформації в Інтернеті, особливо, коли вона потрібна негайно. [1]

Рішення, що існують на сьогоднішній день, не можуть гарантувати достовірність інформації, наданої користувачеві. Багато з онлайн-сервісів — це прості портали чи форуми, де кожен може залишити свій коментар. Але сьогодні є і вузько спеціалізовані сервіси, що націлені на одну тематику.

Сучасні популярні рішення в сфері юриспруденції пропонують автоматизований процес отримання відповіді на питання, що турбує користувача та пошуку спеціаліста, до того ж, це повністю онлайн-рішення. Проте жоден з сервісів ніяк не перевіряє консультантів, що реєструються на сервісах та надають відповіді на питання. Тому виникла ідея створити платформу, де кожного спеціаліста буде валідувати модератор системи. Деякі з них не мають мобільного додатку. А користувачу, що прагне отримати відповідь, в першу чергу потрібна мобільність. Він не має сидіти постійно в Інтернеті, щоб дочекатися відповіді. Для користувача зручніше завантажити додаток, залишити питання та чекати повідомлення.

Метою даної бакалаврської роботи є — спрощення процесу пошуку консультанта та надання користувачу можливості отримати юридичну консультацію в онлайн - режимі.

Завданням даної роботи є створення онлайн - платформи для надання юридичних послуг в частині розробки веб - сервісу для реалізації взаємодії консультанта з платформою, REST API для мобільних додатків, що реалізують взаємодію клієнта з даною платформою, адміністративної панелі для керування платформою та інтегрування зовнішніх сервісів для підтримки дієздатності платформи.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Результатом роботи є розроблена платформа для надання юридичних послуг в онлайн - режимі.

					КПІ.ІП-5121.045440.01.81	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Онлайн-платформа для надання юридичних послуг "Консультант" надає можливість консультації з юристом, адвокатом, бухгалтером в онлайн-режимі. Основна функція і завдання онлайн-платформи - зв'язати клієнта і юриста / адвоката / бухгалтера для надання допомоги клієнту при виникненні будь-яких питань, пов'язаних з юриспруденцією.

Клієнт може задати будь-яке питання і отримати на нього відповідь від кількох виконавців, зареєстрованих в базі. Допускається необмежена кількість відповідей на питання клієнта, тому клієнт зобов'язаний вибрати одну з них, внаслідок чого у нього спишеться певна сума з банківської картки, яку він прив'язує при реєстрації на сервісі. Якщо клієнт не вибере відповідь, він не може ставити наступне питання.

1.2 Змістовний опис і аналіз предметної області

«Якщо вас немає в Інтернеті - значить ви не існуєте» — вираз став крилатим. Сьогодні будь-яка юридична особа, комерційна структура чи підприємство формально починає і здійснює свою діяльність з урахуванням своєї присутності в Інтернеті. Однак закони, укази, постанови, положення тощо змінюються, набувають чинності, скасовуються мало не щодня. Відстежувати цей процес пересічному підприємцю непросто, а іноді й дуже важко, адже йому доведеться витратити величезну кількість часу на дослідження і вирішення тих питань, які безпосередньо з його діяльністю не пов'язані.[3]

На сьогоднішній день є дуже актуальною проблема пошуку юриста чи бухгалтера. Оскільки слідкувати за подіями у змінах законах для людини з іншою сфери є дуже проблематичним, а інколи і нереальним. Проте виникають події, коли допомога спеціаліста є необхідною. Тому виникла ідея створення

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

онлайн-сервісів, які дозволяють знайти консультанта без втрати часу в пошуку одного серед безлічі сайтів та нескінченних телефонних дзвінків.

1.3 Аналіз успішних ІТ-проектів

1.3.1 Аналіз відомих технічних рішень

На сьогоднішній день існує не велика кількість рішень для отримання консультацій по юридичним питанням та пошуку фахівця онлайн. Існує безліч корпоративних сайтів різних юридичних компаній та особистих сайтів юристів, адвокатів, бухгалтерів тощо. Та для звичайного користувача є дуже незручним шукати серед безлічі сайтів гідного кандидата, писати кожному з них для консультації.

Проте сьогодні вже існують ІТ-рішення, які пропонують користувачу залишити свою проблему онлайн та очікувати на відповіді від фахівців. Більшість сервісів мають однаковий функціонал та відрізняються деякими особливостями.

Такі застосування пропонують такі можливості, як наприклад:

- залишити своє повідомлення на сайті та збирати коментарі від різних людей зареєстрованих у системі, в тому числі і справжніх фахівців;
- переглядати список усіх зареєстрованих спеціалістів, перерахувати їх профіль та опис та обрати підходящого для себе;
- перегляд вже існуючих рішень, з якими стикалися раніше. Проте великих недолік такого порталу у тому що закони змінюються, а отже і рішення, яке було актуально раніше вже не актуально на сьогоднішній день.

1.3.2 Аналіз відомих програмних продуктів

Найбільш популярні сервіси надання юридичних послуг:

kabanchik.ua (<https://kabanchik.ua>)

Найбільш відомий сервіс з існуючих рішень у галузі пошуку виконавців будь якої роботи. Сервіс реалізований під Веб, iOS та Android.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Система є спроектованою зручно для користувача, має зручний графічний інтерфейс та можливість вбудовувати опитування.

Недоліком такого сервісу є відсутність перевірки виконавців, немає націленої тематики сервісу, низька кількість зареєстрованих юристів.

freelawyer.ua (<https://www.freelawyer.ua>)

Цей інструмент є потужним та більш підходящим для пошуку консультантів.

Перевагою даного сервісу є каталог зареєстрованих консультантів в відкритій формі (щоб відкрити контакти консультанта потрібно оплатити його вартість), можливість купити на сайті готові документи, можливість пошуку готових рішень на сайті, проте також за окрему плату.

Недоліком системи є відсутність перевірки документів консультантів, те, що на одне питання можуть відповісти максимум 30 консультантів та відсутність мобільних додатків.

KPMG [7]

KPMG – це міжнародна мережа фірм, що надає аудиторські, податкові, юридичні та консультаційні послуги. Недоліками цього сервісу є відсутність мобільних додатків, незручний та складний інтерфейс, неможливість зареєструватись як консультант, надає допомогу тільки компаніям.

protocol.ua

Вказаний інструмент надає можливість тільки переглянути каталог юристів та цікаві судові рішення. Даний сервіс нагадує більше БЛОГ, аніж онлайн-рішення для пошуку спеціалістів чи надання консультацій.

lawyer.ua

Найбільший та по своїм можливостям найпотужніший веб-сервіс на ринку України по наданню юридичних послуг. Недоліки: в системі можуть зареєструватися тільки адвокат або юрист, немає мобільних додатків для клієнта, немає перевірки документів і статусу спеціалістів, вартість послуг залежить від обсягу роботи і кваліфікації фахівця.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

prostopravo.com.ua

Надає можливість пошуку зареєстрованих консультантів. Веб-сервіс реалізований у формі порталу з описом прав громадян і бізнесу. Недоліками сервісу є: онлайн-консультації можливі тільки як особисте повідомлення одному з юристів, як і у всіх попередніх сервісах відсутня перевірка консультантів, відсутність мобільних додатків.

1.4 Аналіз вимог до програмного забезпечення

Для визначення вимог до програмного забезпечення необхідно визначити ролі користувачів та їх можливості в системі. Для досягнення мети розробки система повинна містити наступні типи користувачів:

- клієнт;
- консультант;
- адміністратор;
- модератор.

Консультант включає в себе 5 типів користувача: Юрист, Юридична компанія, Адвокат, Бухгалтер та Бухгалтерська компанія. Всі перераховані ролі мають однаковий функціонал, проте кожен з них може відповідати на питання тільки його тематики.

Через мобільний додаток з сервісом буде взаємодіяти винятково клієнт. Виконавець буде працювати через веб-ресурс.

а) Додаток зв'язується з веб-ресурсом через API. Всі питання, які створює клієнт в додатку, через API синхронізуються з веб-ресурсом, що дозволяє їх бачити і відповідати на них виконавцю.

б) Щоб поставити запитання, клієнт скачує мобільний додаток, реєструється, після чого отримує доступ до всього функціоналу, банківську карту можна прив'язати пізніше.

в) Стартова сторінка додатка після авторизації / реєстрації - відкривається сторінка "Задати питання".

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

г) При першій авторизації додаток має в примусовому порядку запитувати дозвіл на включення Push повідомлень.

д) Після входу, додаток повинен завжди працювати в автономному режимі, тобто якщо його просто закрити без натискання на кнопку "Вихід", при наступному запуску не потрібно знову вводити логін і пароль

е) У додатку передбачена програма лояльності (реферальна система).

є) Вихід з особистого кабінету здійснюється за допомогою натискання на кнопку "Вихід".

ж) При бездіяльності користувача (коли він не відкриває додаток протягом певного часу, не заходить в нього) або при певних діях додаток повинен показувати користувачеві запрограмовані заздалегідь на певні умови push повідомлення. Повинно бути запрограмовані не менше 40 повідомлень при виконанні різних умов. Повідомлення не можливо змінювати з адміністративної панелі.

Структура веб-ресурсу:

Відкрита частина сайту (для всіх користувачів):

- а) реєстрація, авторизація та відновлення пароля;
- б) головна сторінка;
- в) сторінка "Контакти";
- г) сторінки "Угода" і "Політика конфіденційності".

Видима частина сайту тільки для авторизованих користувачів - особистий кабінет виконавця. (Юрист / адвокат / юридична фірма / бухгалтер / бухгалтерська фірма):

- а) актуальні питання;
- б) архів питань (на які були дані відповіді);
- в) мій профіль;
- г) додати карту.

Адміністративна панель:

- авторизація, відновлення пароля;

- адміністратор:

а) список виконавців, які подали заявку на реєстрацію;

б) список виконавців, які пройшли модерацію;

в) список виконавців, які не пройшли модерацію;

г) список заблокованих виконавців;

д) список клієнтів;

е) список модераторів;

є) налаштування;

- модератор.

1.4.1 Розроблення функціональних вимог

В системі передбачено наступні варіанти використання, які наведені у таблицях 1.1. – 1.22.

Таблиця 1.1 – Варіант використання UC001

Назва	Авторизація адміністратора
Опис	Адміністратор має можливість авторизуватись в системі.
Учасники	Адміністратор.
Передумови	
Постумови	Адміністратор проходить авторизацію у веб-сервісі.

Продовження таблиці 1.1

Основний сценарій	Система демонструє вікно авторизації, яке містить адрес електронної пошти та пароль. Адміністратор заповнює поля вводу.. Система демонструє кнопку “Увійти”. Адміністратор натискає кнопку “Увійти”. Система авторизує адміністратора.
Розширення сценаріїв	4.1. Система виявляє, що дані, внесені адміністратором, не є валідними. 4.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.

Таблиця 1.2 – Варіант використання UC002

Назва	Авторизація модератора
Опис	Модератор має можливість авторизуватись в системі.
Учасники	Модератор.
Постумови	Модератор проходить авторизацію у Веб-сервісі
Основний сценарій	Система демонструє вікно авторизації, яке містить адрес електронної пошти та пароль. Модератор заповнює поля вводу.. Система демонструє кнопку “Увійти”. Модератор натискає кнопку “Увійти”. Система авторизує модератора.

Продовження таблиці 1.2

Розширення сценаріїв	<p>4.1. Система виявляє, що дані, внесені модератором, не є валідними.</p> <p>4.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.</p>
----------------------	--

Таблиця 1.3 – Варіант використання UC003

Назва	Реєстрація нового юзера
Опис	Адміністратор має можливість створити нового юзера.
Учасники	Адміністратор.
Передумови	Користувач авторизований як адміністратор
Основний сценарій	<p>1. Система демонструє вікно для створення нового користувача, консультанта, модератора</p> <p>2. Адміністратор заповнює необхідні поля для реєстрації юзера (ПІБ, номер телефону, електронна пошта, пароль) та обирає роль нового користувача(клієнт, консультант, модератор)</p> <p>Система демонструє кнопку “Зберегти”.</p> <p>Адміністратор натискає кнопку “Зберегти”.</p> <p>Система створює нового користувача.</p>
Розширення сценаріїв	<p>4.1. Система виявляє, що дані, внесені адміністратором, не є валідними.</p> <p>4.1.a. Система демонструє користувачеві повідомлення про відповідну помилку.</p>

Таблиця 1.4 – Варіант використання UC004

Назва	Валідація нового консультанта.
Опис	Адміністратор, модератор мають можливість підтвердити валідність нового консультанта.
Учасники	Адміністратор, модератор.
Передумови	Користувач авторизований як Адміністратор або Модератор.
Постумови	Консультант активний у системі та має можливість відповідати клієнтам.
Основний сценарій	<p>Адміністратор або модератор переглядають новозареєстрованих консультантів.</p> <p>Адміністратор або модератор переглядають інформацію про нового консультанта та завантажують документи, що прикріпив консультант.</p> <p>Якщо усі документи валідні, адміністратор або модератор підтверджують консультанта.</p> <p>Адміністратор чи модератор натискає на кнопку "Підтвердити".</p> <p>Система надає консультанту статус «Активний».</p>
Розширення сценаріїв	Якщо документи консультанта або інформація про нього не задовольняють умовам його блокують у системі

Таблиця 1.5 – Варіант використання UC005

Назва	Зміна інформації консультанта
Опис	Адміністратор або модератор мають можливість переглянути запит консультанта на зміну інформації чи роду діяльності та погодити зміну.
Учасники	Адміністратор або модератор.
Передумови	Користувач авторизований як Адміністратор або Модератор.
Постумови	Зміна інформації або роду діяльності консультанта
Основний сценарій	<p>Адміністратор або модератор переглядає консультантів, що бажають поміняти інформацію чи змінити діяльність.</p> <p>Адміністратор або модератор переглядають стару інформацію про консультанта, нову інформацію, що просить консультант та завантажують документи, що прикріпив консультант.</p> <p>Якщо усі документи валідні, адміністратор або модератор підтверджують зміни консультанта.</p> <p>Адміністратор чи модератор натискає на кнопку “Підтвердити”.</p> <p>Система змінює інформацію про консультанта.</p>
Розширення сценаріїв	Якщо документи консультанта або інформація про нього не задовольняють умовам його зміни не підтверджуються

Таблиця 1.6 – Варіант використання UC006

Назва	Реєстрація нового клієнта
Опис	Клієнт має можливість зареєструватись у системі через мобільний додаток
Учасники	Користувач.
Передумови	-
Постумови	Новозареєстрований клієнт
Основний сценарій	Користувач заповнює дані реєстраційної форми. Користувач натискає кнопку “зареєструватись”. Система надсилає смс-повідомлення з кодом аутентифікації Користувач надсилає 6-значний код. Система аутентифікує клієнта та перенаправляє до профілю користувача.
Розширення сценаріїв	

Таблиця 1.7 – Варіант використання UC007

Назва	Реєстрація клієнта через реферальне посилання
Опис	Користувач реєструється з реферальним посиланням на сайті.
Учасники	Користувач

Продовження таблиці 1.7

Передумови	-
Постумови	Новозареєстрований клієнт, як реферал існуючого клієнта у системі.
Основний сценарій	<p>Користувач переходить за посиланням на веб сайт з унікальним токеном реферала</p> <p>Користувач заповнює дані реєстраційної форми</p> <p>Користувач натискає кнопку “зареєструватись”.</p> <p>Система надсилає смс-повідомлення з кодом аутентифікації</p> <p>Користувач надсилає 6-значний код.</p> <p>Система реєструє клієнта</p>
Розширення сценаріїв	

Таблиця 1.8 – Варіант використання UC008

Назва	Зміна інформації клієнта в профілі
Опис	Користувач має можливість змінити інформацію про себе в профілі
Учасники	Користувач.
Передумови	Користувач авторизований як Клієнт.
Постумови	Профіль користувача з новими даними

Продовження таблиці 1.8

Основний сценарій	Користувач заповнює поля, запропоновані в формі його профілю Користувач натискає кнопку «Зберегти» Дані форми надходять на сервер та зберігаються у БД
Розширення сценаріїв	Якщо дані, введені користувачем не є валідні, сервер відправляє повідомлення з помилкою

Таблиця 1.9 – Варіант використання UC009

Назва	Залишити запитання консультантам
Опис	Користувач має можливість залишити питання на сервісі
Учасники	Користувач.
Передумови	Користувач авторизований як Клієнт.
Постумови	Збережене питання на сервісі
Основний сценарій	Користувач заповнює дані форми Дані відправляються на сервіс та зберігаються у БД З користувача списуються гроші з картки, або зменшується кількість безкоштовних питань
Розширення сценаріїв	Якщо у клієнта немає безкоштовних питань та не прив'язана карта, користувачу висвітлюється відповідне повідомлення

Таблиця 1.10 – Варіант використання UC010

Назва	Переглянути відповіді на питання.
Опис	Користувач має можливість переглянути відповіді консультантів на його питання
Учасники	Користувач.
Передумови	Користувач авторизований як Клієнт, що створив питання
Постумови	Висвітлені користувачу відповіді на питання
Основний сценарій	Користувач вибирає питання, на яке хоче переглянути відповіді Користувачу у вікні виводяться усі відповіді консультантів
Розширення сценаріїв	

Таблиця 1.11 – Варіант використання UC011

Назва	Обрати найкращу відповідь.
Опис	Користувач може обрати найкращу відповідь із запропонованих
Учасники	Користувач
Передумови	Користувач, авторизований як Клієнт, що створив питання та отримав відповіді на нього

Продовження таблиці 1.11

Постумови	Питання набуває статусу «Закритий»
Основний сценарій	Користувач переглядає усі відповіді на питання Користувач натискає на одне з відповідей, що є найбільш задовільним Користувач натискає кнопку «Обрати як найкраще»
Розширення сценаріїв	

Таблиця 1.12 – Варіант використання UC012

Назва	Відкрити дані консультанта.
Опис	Користувач має можливість відкрити дані консультанта, що залишив відповідь на його питання.
Учасники	Користувач
Передумови	Користувач, авторизований як Клієнт, створив питання і отримав відповідь у ньому.
Постумови	Відкритий профіль консультанта.
Основний сценарій	Користувач натискає на створене питання. Користувач переглядає усі відповіді Користувач обирає одну з відповідей Користувач натискає на кнопку «Відкрити дані консультанта» З користувача знімається оплата за відкриття даних консультанта.

Продовження таблиці 1.12

Розширення сценаріїв	Якщо у клієнта не прив'язана карта, користувачу висвітлюється відповідне повідомлення
----------------------	---

Таблиця 1.13 – Варіант використання UC013

Назва	Налаштування оповіщень.
Опис	Користувач може ввімкнути або вимкнути оповіщення.
Учасники	Користувач
Передумови	Користувач, авторизований як Клієнт.
Постумови	Ввімкнені або вимкнуті оповіщення
Основний сценарій	Користувач натискає на вкладку налаштування та обирає оповіщення Користувач може пересунути повзунок оповіщень та ввімкнути чи вимкнути Push-повідомлення.
Розширення сценаріїв	

Таблиця 1.14 – Варіант використання UC014

Назва	Додати кредитну картку.
Опис	Користувач має можливість додати кредитну картку.
Учасники	Користувач.
Передумови	Користувач, авторизований як Клієнт.
Постумови	Картка додана до системи
Основний сценарій	Користувач натискає на вкладку налаштування та обирає функцію «додати картку» Користувач вводить дані картки до форми у вікні Користувач натискає кнопку «Зберегти» Дані картки зберігаються у системі
Розширення сценаріїв	Якщо дані картки не є валідні та API wayforpay надсилає повідомлення про невірні дані користувачу висвітлюється відповідне повідомлення

Таблиця 1.15 – Варіант використання UC015

Назва	Змінити телефон акаунта
Опис	Користувач має можливість змінити номер телефону акаунта.

Продовження таблиці 1.15

Учасники	Користувач
Передумови	Користувач, що попередньо зареєструвався та ввів в профіль свою електронну пошту та пароль
Постумови	Користувач змінює номер телефону
Основний сценарій	<p>Користувач натискає кнопку змінити телефон</p> <p>Користувач вводить номер телефону на електронну пошту</p> <p>Користувачу на пошту приходить посилання з унікальним токеном</p> <p>Користувач на сайті вводить дані, а саме: електронну пошту, старий номер телефону, пароль та новий номер телефону</p> <p>Якщо всі дані вірні, в БД записується новий номер телефону, а старий видаляється</p>
Розширення сценаріїв	4.1. Якщо одні з даних не є вірні, а саме, пошта, старий номер телефону, пароль або токен, що прив'язаний до посилання, користувачу висвітлюється відповідне повідомлення

Таблиця 1.16 – Варіант використання UC016

Назва	Реєстрація нового консультанта
Опис	Консультант має можливість зареєструватись у системі через веб сервіс
Учасники	Користувач.
Передумови	-
Постумови	Новозареєстрований консультант
Основний сценарій	Користувач заповнює дані реєстраційної форми. Користувач натискає кнопку “зареєструватись”. Система надсилає повідомлення на електрону пошту Користувач підтверджує електронну пошту через перехід по посиланню Система аутентифікує клієнта та перенаправляє до профілю консультанта.
Розширення сценаріїв	

Таблиця 1.17 – Варіант використання UC017

Назва	Реєстрація консультанта через реферальне посилання
Опис	Користувач реєструється з реферальним посиланням на сайті.
Учасники	Користувач

Продовження таблиці 1.17

Передумови	-
Постумови	Новозареєстрований консультант, як реферал існуючого консультанта у системі.
Основний сценарій	<p>Користувач переходить за посиланням на веб- сайт</p> <p>Користувач заповнює дані реєстраційної форми</p> <p>Консультант заповнює у спеціальному полі реферальний номер</p> <p>Користувач натискає кнопку “zareestruvatys”.</p> <p>Система надсилає повідомлення на електронну пошту</p> <p>Користувач підтверджує електронну пошту через перехід по посиланню</p> <p>Система реєструє консультанта</p> <p>Система додає консультанта у список новозареєстрованих у панелі адміністратора для подальшої верифікації</p> <p>Система перенаправляє консультанта до профілю</p>
Розширення сценаріїв	

Таблиця 1.18 – Варіант використання UC018

Назва	Зміна інформації консультанта в профілі
Опис	Консультант має можливість змінити інформацію про себе в профілі
Учасники	Користувач.
Передумови	Користувач авторизований як Консультант.
Постумови	Профіль консультанта з новими даними
Основний сценарій	<p>Консультант заповнює поля, запропоновані в формі його профілю</p> <p>Користувач натискає кнопку «Зберегти»</p> <p>Дані форми надходять на сервер та зберігаються у спеціальній таблиці БД для подальшої верифікації модератором</p>
Розширення сценаріїв	<p>Якщо дані, введені консультантом не валідні, система повідомляє відповідним повідомленням</p> <p>Якщо дані користувача валідні та не проходять верифікацію модератором, дані користувача не зберігаються, та його профіль залишається зі старими даними</p> <p>Якщо дані консультанта валідні та проходить верифікацію модератором, дані консультанта у профілі змінюються автоматично</p>

Таблиця 1.19 – Варіант використання UC019

Назва	Зміна пароля консультанта
Опис	Консультант має можливість змінити свій пароль
Учасники	Користувач
Передумови	Користувач, що зареєстрований в системі
Постумови	Консультант змінює пароль до профілю
Основний сценарій	<p>Користувач натискає на кнопку «Забув пароль»</p> <p>Система перенаправляє користувача на спеціальну форму</p> <p>Користувач заповнює дані форми, а саме свою електронну пошту</p> <p>На пошту консультанту приходить посилання до форми для зміни пароля з унікальним токеном</p> <p>Консультант заповнює дані форми, а саме, свою пошту новий пароль та підтвердження пароля</p> <p>Якщо токен та пошта співпадають, в БД записується новий пароль консультанта</p>
Розширення сценаріїв	Якщо токен або пошта не валідні, користувачу висвітлюється відповідне повідомлення

Таблиця 1.20 – Варіант використання UC020

Назва	Надати відповідь на запитання
Опис	Консультант має можливість залишити відповідь на запитання його тематики
Учасники	Консультант
Передумови	Аутентифікований консультант повинен бути верифікованим модератором.
Постумови	Консультант зберігає свою відповідь на питання у БД
Основний сценарій	<p>Консультант переглядає питання його тематики на Дешборді</p> <p>Консультант обирає питання, яке його цікавить</p> <p>Консультант заповнює поле для відповіді та натискає кнопку «Зберегти»</p> <p>Відповідь консультанта зберігається та питання зникає з вікна дешборда</p> <p>Питання з відповіддю консультанта зберігається у вкладці архіву</p>
Розширення сценаріїв	Якщо консультанта не цікавить питання, він має можливість натиснути кнопку «Не цікавить», та питання зникає з вікна дешборда та потрапляє у стопліст

Таблиця 1.21 – Варіант використання UC021

Назва	Налаштування оповіщень.
Опис	Консультант може ввімкнути або вимкнути оповіщення.
Учасники	Користувач
Передумови	Користувач, авторизований як Консультант.
Постумови	Ввімкнені або вимкнуті оповіщення
Основний сценарій	Консультант натискає на вкладку налаштування та обирає оповіщення Консультант може пересунути повзунок оповіщень та ввімкнути чи вимкнути Сесійні оповіщення.
Розширення сценаріїв	

Таблиця 1.22 – Варіант використання UC022

Назва	Додати кредитну картку.
Опис	Консультант має можливість додати кредитну картку.
Учасники	Користувач.
Передумови	Користувач, авторизований як Консультант.
Постумови	Картка додана до системи

Продовження таблиці 1.22

Основний сценарій	Консультант натискає на вкладку налаштування та обирає функцію «додати картку» Консультант вводить дані картки до форми у вікні Консультант натискає кнопку «Зберегти» Дані картки зберігаються у системі
Розширення сценаріїв	Якщо дані картки не є валідні та API wayforpay надсилає повідомлення про невірні дані консультанту висвітлюється відповідне повідомлення

Схема структурна діяльності в процесі консультування за допомогою платформи наведена на рисунку 1.1

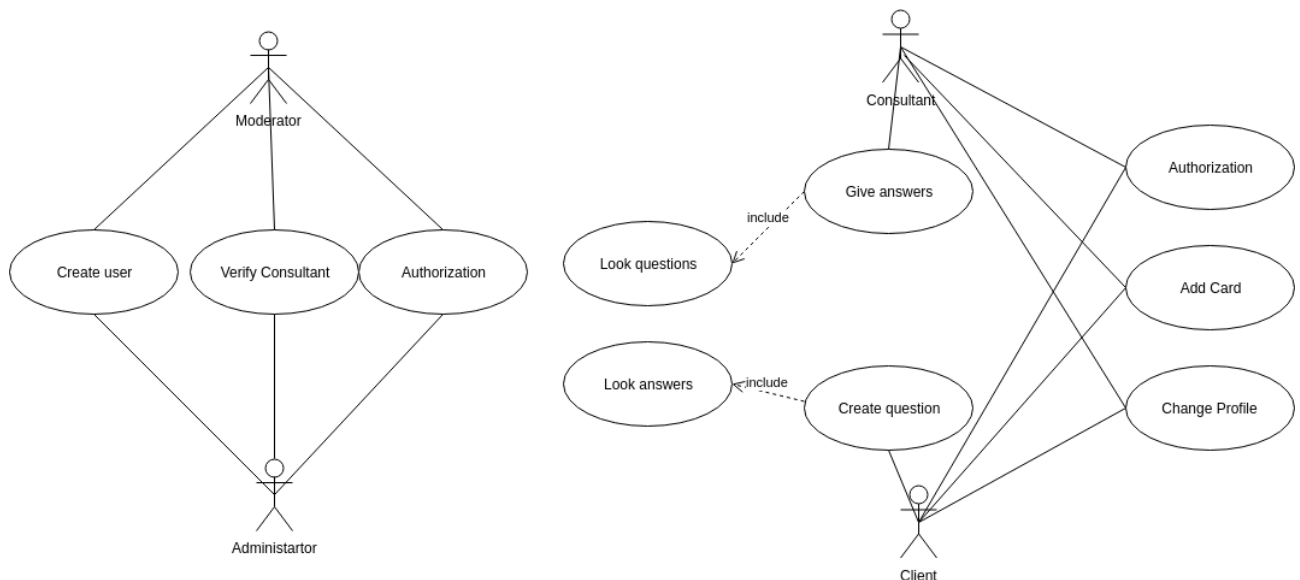


Рисунок 1.1 – Схема структурна діяльності в процесі консультування за допомогою платформи

Залежності між функціональними вимогами клієнтського додатку зображено на рисунку 1.2

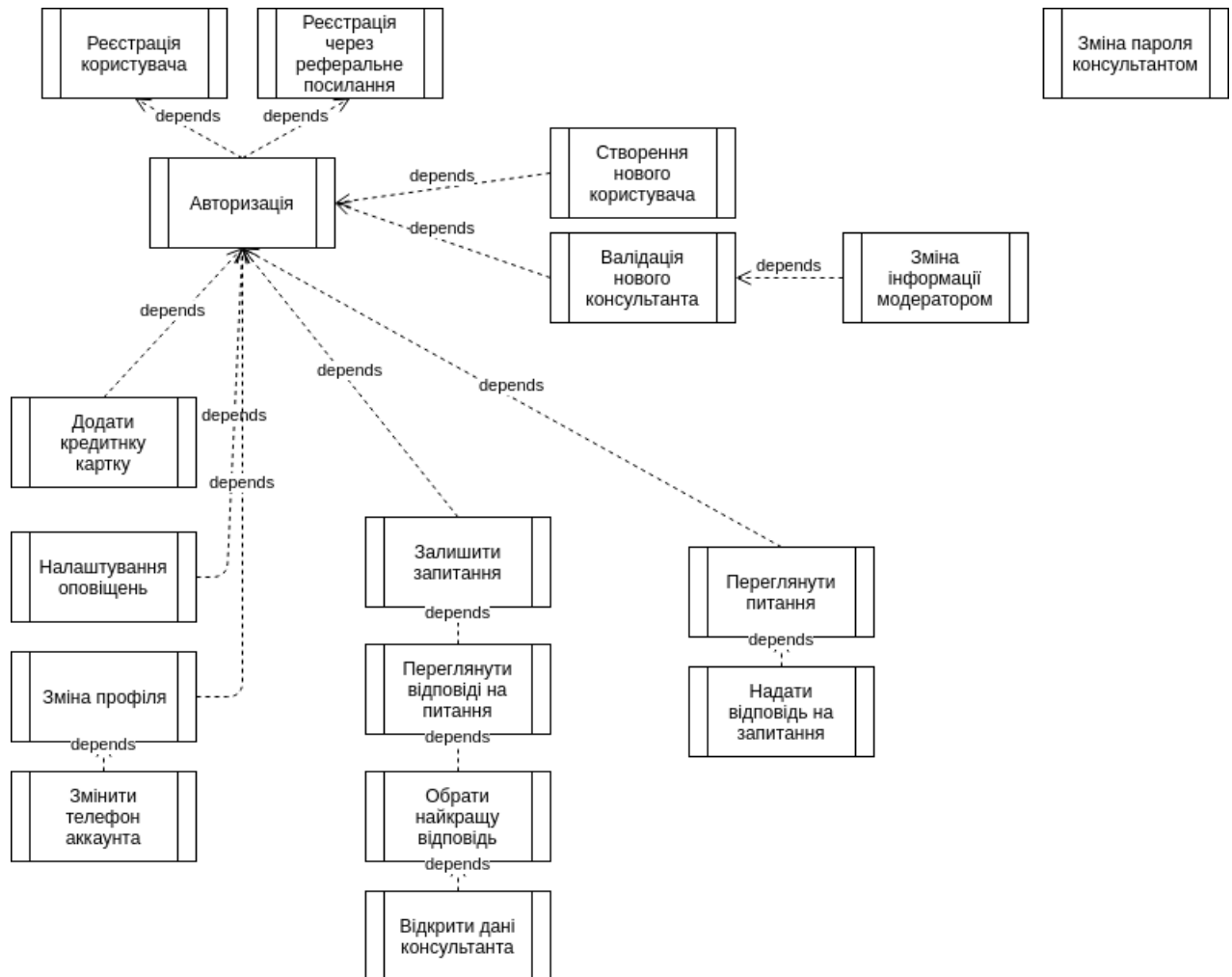


Рисунок 1.2 – Діаграма залежності між функціональними вимогами

Взаємозв'язки між вимогами клієнтського додатку і варіантами використання відображено на рисунку 1.3

	Реєстрація користувача	Реєстрація через реф.посилання	Авторизація	Ствердження нового користувача	Валідація нового консультанта	Зміна інформації модератором	Зміна пароля консультантом	Додати кредитну картку	Налаштування оповіщень	Зміна профіля	Змінити телефон аккаунта	Залишити запитання	Переглянути відповіді на питання	Обрати найкращу відповідь	Відкрити дані консультанта	Переглянути питання	Надати відповідь на запитання
Реєстрація користувача	■																
Реєстрація через реф.посилання		■															
Авторизація	■	■	■														
Ствердження нового користувача			■	■													
Валідація нового консультанта	■	■	■		■												
Зміна інформації модератором	■	■	■		■	■											
Зміна пароля консультантом	■	■					■										
Додати кредитну картку	■	■	■		■			■									
Налаштування оповіщень	■	■	■						■								
Зміна профіля	■	■	■		■					■							
Змінити телефон аккаунта	■	■			■						■						
Залишити запитання	■	■	■					■				■					
Переглянути відповіді на питання	■	■	■					■				■	■				
Обрати найкращу відповідь	■	■	■					■				■	■	■			
Відкрити дані консультанта	■	■	■					■				■	■		■		
Переглянути питання	■	■	■		■											■	
Надати відповідь на запитання	■	■	■		■											■	■

Рисунок 1.3 – Матриця залежності між вимогами додатку і варіантами використання

1.4.2 Розроблення нефункціональних вимог

Програмне забезпечення повинне відповідати наступним нефункціональним вимогам:

- локалізація інтерфейсу – українська для української локалізації пристрою;

- підтримувана версія ОС для мобільних додатків Android 4.4 KitKat та вище;
- підтримувана версія ОС для мобільних додатків iOS 9 та вище;
- клієнт-серверна взаємодія повинна бути реалізована у вигляді REST API для мобільних платформ;
- для передачі даних повинні використовуватись захищені канали передачі даних з використанням TLS;
- використання сесійного режиму;
- зберігання важливих даних користувача на веб-сервісу в зашифрованому вигляді;
- підтримка усіх розширень екрану;
- підтримка усіма сучасними браузерами.

1.4.3 Постановка комплексу завдань

Призначенням розробки є створення онлайн - платформи для надання юридичних послуг в частині розробки веб-сервісу для реалізації взаємодії консультанта з платформою, REST API для мобільних додатків, що реалізують взаємодію клієнта з даною платформою, адміністративної панелі для керування платформою та інтегрування зовнішніх сервісів для підтримки дієздатності платформи.

Мета створення даної роботи – спрощення процесу пошуку консультанта та надання користувачу можливості отримати юридичну консультацію в онлайн – режимі.

Для досягнення поставлених цілей необхідно вирішити наступні задачі:

- розробити веб-сервіс для консультантів;
- розробити REST API для мобільних платформ;
- розробити адміністративну панель;
- інтегрувати REST та SOAP API сторонніх сервісів.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Для досягнення мети даної роботи онлайн - платформа повинна мати наступний функціонал.

Для користувача:

- формування питання (тип, саме питання, тощо);
- перегляд всіх заданих користувачем питань;
- перегляд всіх відповідей різних консультантів на поставлене питання;
- можливість відкрити контакти потрібного консультанта.

Для консультанта:

- надання відповіді на питання;
- перегляд списку усіх відповідей користувачам;
- перегляд всіх питань, на які ще не надані відповіді;
- завантаження своїх документів для верифікації при реєстрації на платформі.

Для модератора:

- перегляд списку консультантів, зареєстрованих на платформі;
- валідація новозареєстрованих консультантів та тих, що змінюють свої дані.

Для адміністратора:

- перегляд списку консультантів, зареєстрованих на платформі;
- валідація новозареєстрованих консультантів та тих, що змінюють свої дані;
- створення нового користувача у системі;
- блокування користувача у системі.

Мобільний додаток повинен працювати на пристроях зі встановленою ОС Android та iOS. Має бути присутній Веб-сервіс для консультантів та адміністративна панель для модераторів і адміністраторів. Бізнес-логіка має бути реалізована на серверній частині додатку. Керування користувачами має здійснюватися з адміністративної веб-панелі додатку.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

1.5 Висновки по розділу

Існує ряд програмних рішень для пошуку консультантів у сфері юриспруденції, однак жодна з переглянутих систем не надає користувачеві гарантії, що консультант має відповідні знання у сфері його тематики. Також, більшість існуючих рішень не мають мобільного додатку для користувача.

Розробка онлайн - платформи, яка містила б функціональність, зазначену вище, є достатньо актуальною. Додатковим фактором, який вказує на зручність використання подібної платформи, є використання саме мобільного додатку для користувача, адже це надає користувачу гнучкість, можливість отримувати повідомлення з системи і не прив'язує до браузерів.

В даному розділі наведено опис предметного середовища, наведена та описана схема структурна діяльності в процесі консультування за допомогою платформи.

Проведено огляд та аналіз функціональності існуючого програмного забезпечення для вирішення зазначених задач та виділені їх недоліки, а саме відсутність підтримки мобільних платформ, верифікації консультантів.

Розроблені функціональні вимоги для кожного з акторів, визначених в процесі бізнес-аналізу та нефункціональних вимог для платформи.

Наведені призначення, мета та задачі розробки.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

2.1.1 Частина користувача

Для створення якісного програмного забезпечення необхідно попереднє детальне моделювання процесів і архітектури програмного забезпечення.

Схема містить загальні процеси, які проходить користувач під час використання додатку. Загальні процеси, які проходить користувач, включають процеси авторизації, реєстрації, зміна профілю та даних користувача, роботу з питаннями та відповідями, збереження картки, керування сповіщеннями, перегляд консультантів. Додатковими є процес зміни номеру телефону користувача.

Послідовний опис етапу реєстрації до додатку:

Користувач заповнює дані реєстраційної форми. Відповіді користувача відправляються на сервер. У разі успішної реєстрації (дані реєстрації, введені користувачем, збереглися в базі даних серверної частини, а також немає акаунта, зареєстрованого з таким же телефоном) користувачу приходить смс-повідомлення з кодом.

Для активування профілю користувач повинен ввести 6-значний код в поле. У разі правильного введення коду акаунт користувача стає активний та підтвердженням.

Користувач отримує спеціальний JWT-токен. З даним токеном користувач може виконувати будь які дії у додатку.

Схема реєстрації до додатку зображена на рисунку 2.1

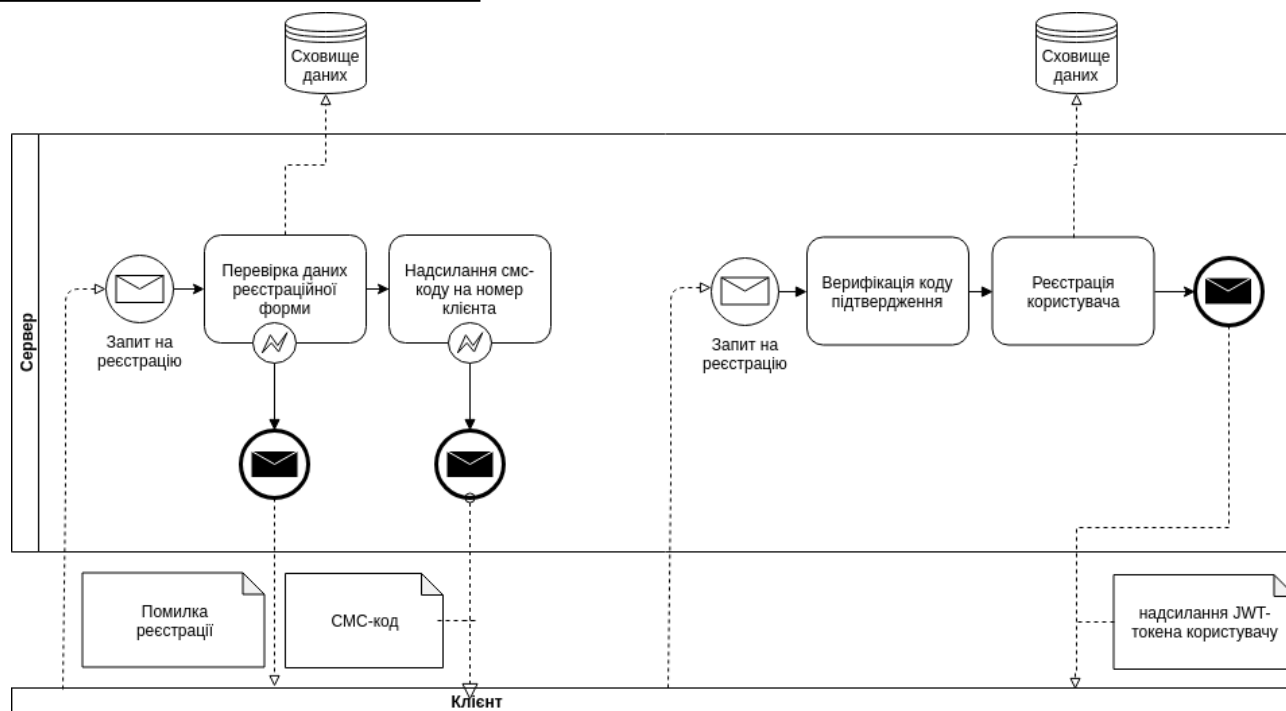


Рисунок 2.1 – Схема структурна бізнес-процесу реєстрації до додатку

Послідовний опис роботи користувача з профілем:

- користувачу демонструється його профіль та перші дані, що ввів при реєстрації;
- користувач має можливість ввести чи змінити будь які поля продемонстровані в профілі;
- після того як користувач ввів всі дані та загрузив своє фото профілю, він має можливість натиснути кнопку “Зберегти”;
- у разі успішного збереження (якщо всі дані збереглися у БД) після прокрутки колеса очікування всі дані відобразяться у користувача на сторінці. У разі неуспішного збереження даних користувачу буде показано відповідне повідомлення у діалоговому вікні.

Схема роботи користувача з профілем зображена на рисунку 2.2

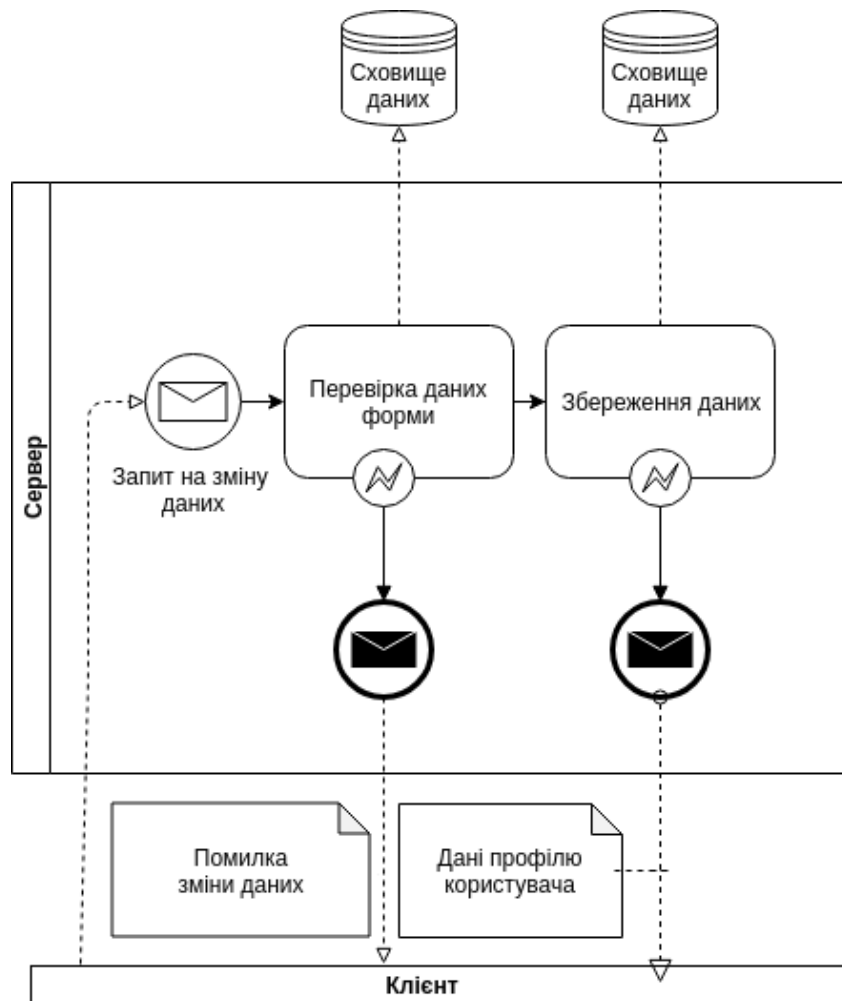


Рисунок 2.2 - Схема структурна бізнес-процесу роботи користувача з профілем.

Послідовний опис роботи користувача з повідомленнями:

- користувач натискає на плюс внизу додатку;
- користувач має можливість ввести заголовок свого питання, розписати питання та обрати виконавця, який буде займатися його питанням (юрист, адвокат, бухгалтер, юридична компанія чи бухгалтерська компанія);
- коли користувач ввів усі поля, він натискає на кнопку “Відправити”;
- якщо в користувача є безкоштовні повідомлення, то кількість безкоштовних повідомлень зменшується, а питання зберігається у БД. Якщо безкоштовних повідомлень немає, та в користувача не прив'язана картка до профілю, тоді користувачу висвітлюється повідомлення з проханням прив'язати

картку до профілю. Якщо в користувача прив'язана картка та немає безкоштовних повідомлень, тоді повідомлення зберігається у БД.

Схема роботи користувача з повідомленнями зображена на рисунку 2.3

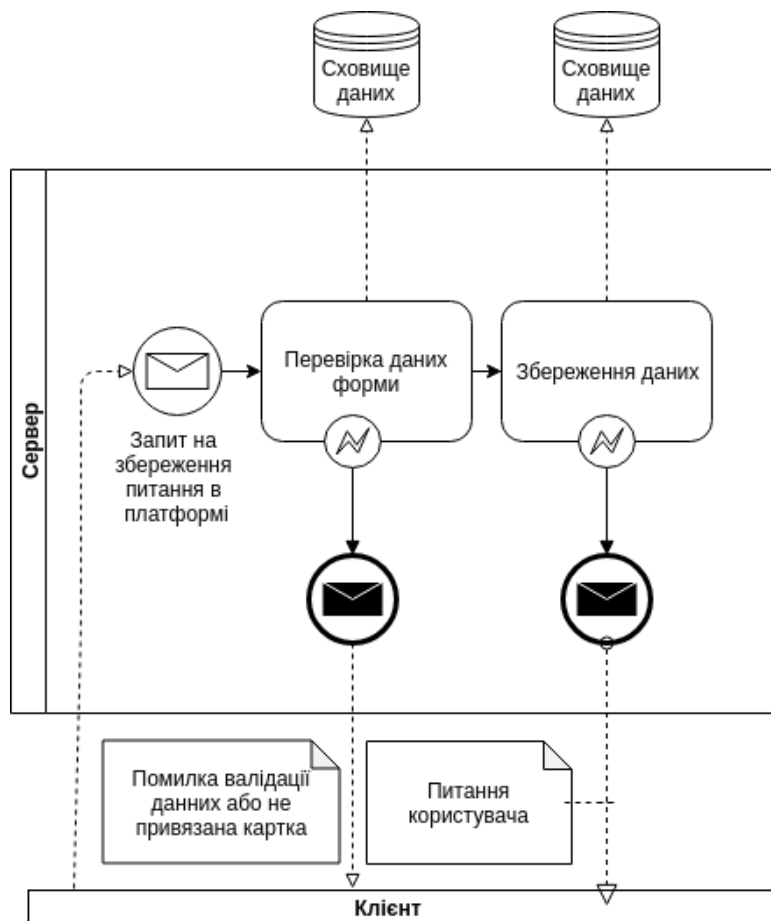


Рисунок 2.3 - Схема структурна бізнес-процесу роботи користувача з повідомленнями

Послідовний опис роботи користувача з оповіщеннями:

- користувач натискає на кнопку налаштувань та обирає вкладку оповіщення;
- користувач має можливість пересунути повзунок налаштувань щодо оповіщень системних та оповіщень нових відповідей на його питання;
- якщо повзунок пересунутий вправо, тоді функція активована, якщо

повзунок пересунутий вліво та не підсвічений синім кольором, тоді функція вимкнена.

Схема роботи користувача з оповіщеннями зображена на рисунку 2.4

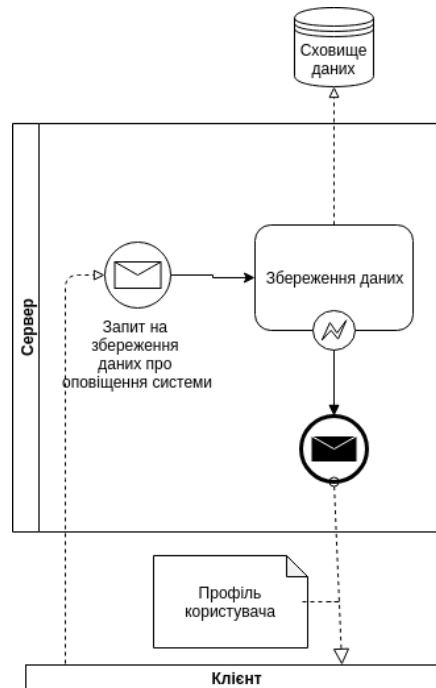


Рисунок 2.4 - Схема структурна бізнес-процесу роботи користувача з оповіщеннями

Послідовний опис роботи користувача з підключенням карти:

- користувач натискає на кнопку налаштувань та обирає вкладку картки;
- користувач має можливість ввести дані картки у поле та натиснути “Зберегти”;
- якщо усі дані валідні та система wayforpay приймає дані картки, тоді дані картки зберігається у БД, а користувачу висвітлюється його дані. Якщо виникає помилка або картка не валідна, тоді користувачу висвітлюється повідомлення у діалоговому вікні.

Схема роботи користувача з підключенням карти зображена на рисунку 2.5

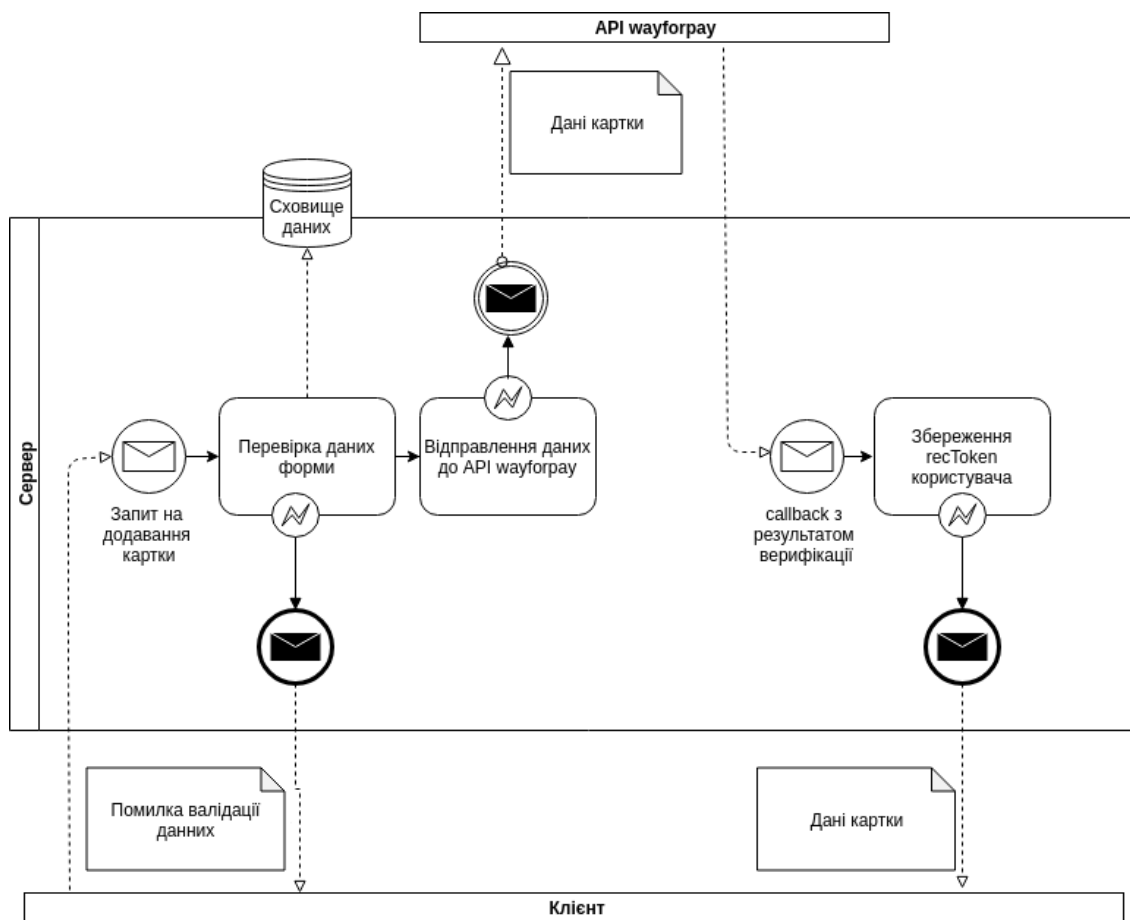


Рисунок 2.5 - Схема структурна бізнес-процесу роботи користувача з підключенням карти

Додатковим сценарієм є процес зміни номеру телефону.

Послідовний опис процесу зміни номеру телефону:

- для зміни номера телефону користувач попередньо мав заповнити поля email та пароль у профілі;
- користувач вводить свою пошту для того, щоб на неї відіслали посилання з спеціальним токена юзера;
- користувачу приходить повідомлення з посиланням на веб-версію форми зміни номеру телефону;
- у формі користувач повинен ввести свою пошту, пароль, що вводив у мобільному додатку, старий номер телефону та новий.

Якщо всі дані вірні, та токен, що був прив'язаний до посилання на веб-форму, відповідає користувачу, тоді номер телефону користувача змінюється на новий. Якщо одні з даних не валідні, тоді користувачу висвітлюється відповідне повідомлення.

2.1.2. Частина консультанта

Послідовний опис процесів для реєстрації консультанта:

- консультант заповнює дані реєстраційної форми та прикріплює свідоцтво та інші документи;
- на пошту консультанту приходить повідомлення з посиланням для верифікації імейлу, а консультанта автоматично переводить на його профіль, де в профілі помітка про неактивний акаунт;
- коли консультант підтвердить свою пошту, його статус зміниться на «не перевірений модератором» і його акаунт потрапить у панель модератора як новозареєстрований;
- коли модератор перевірить документи консультанта і провалідує його, модератор змінить статус консультанта на «активний»;
- коли у консультанта «активний» статус він може відповідати на питання користувачів на вкладці «Dashboard».

Схема реєстрації консультанта зображена на рисунку 2.6

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

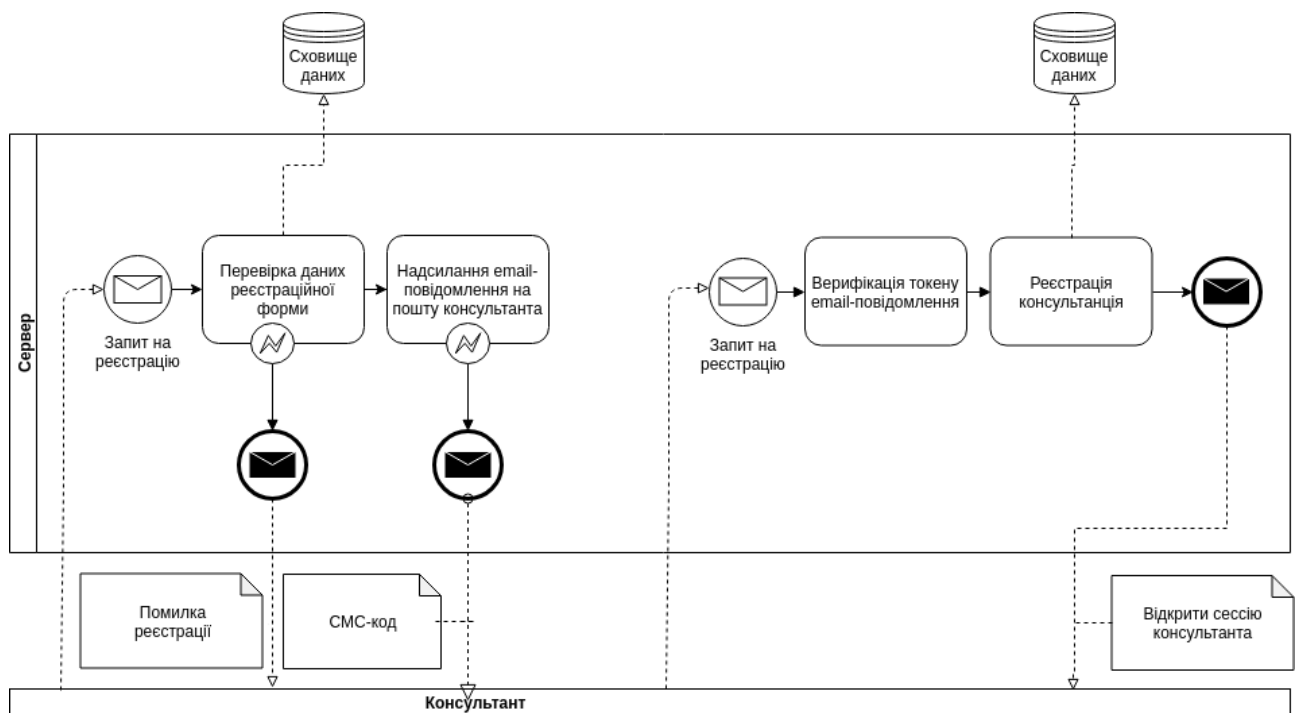


Рисунок 2.6 – Схема структурна бізнес-процесу реєстрації консультанта

Послідовний опис процесів для авторизації консультанта:

- консультант заповнює дані авторизаційної форми та відправляє на сервер;
- сервер валідує дані отримані з фронтенду, перевіряє чи існує такий юзер у системі та перевіряє співпадіння пароля з електронною поштою;
- якщо електронна пошта та пароль співпадають, тоді консультант отримує токен, авторизується у системі та автоматично направляється в профіль.

Схема авторизації консультанта зображена на рисунку 2.7

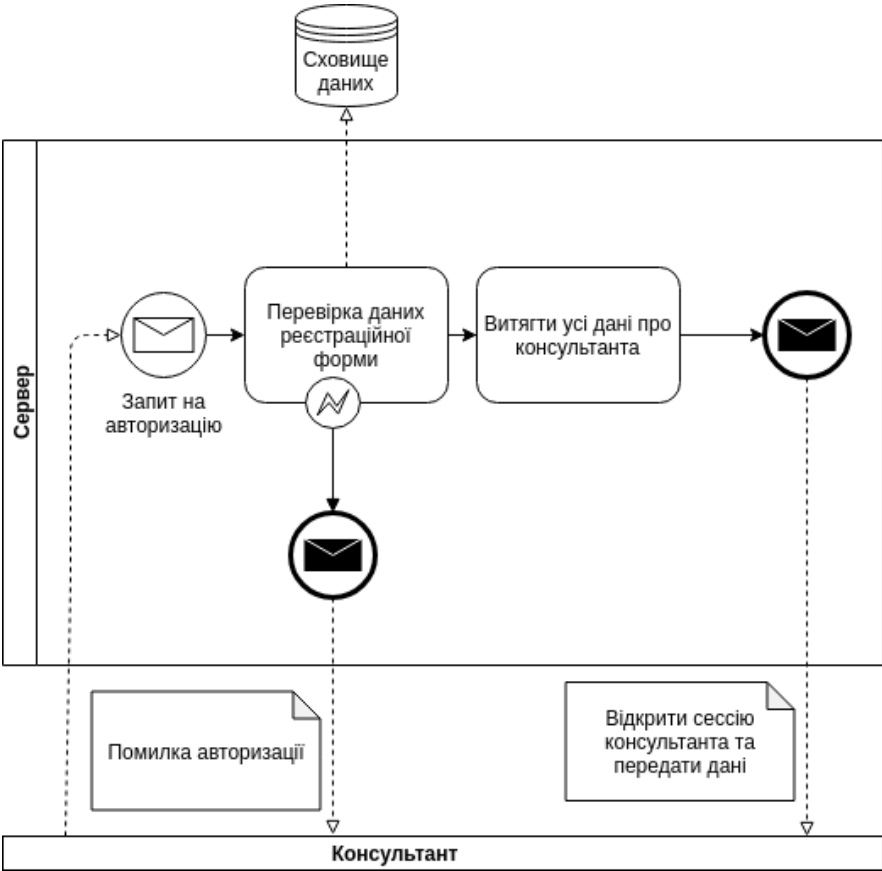


Рисунок 2.7 - Схема структурна бізнес-процесу авторизації консультанта

Схема процесу надання відповідей на питання користувачів зображена на
рисунку 2.8

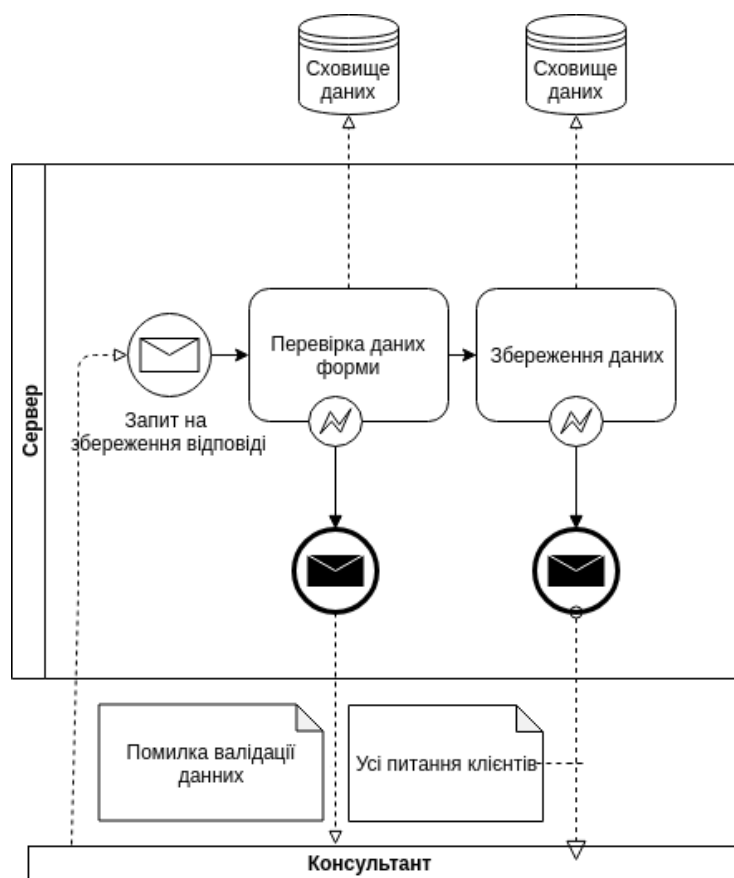


Рисунок 2.8 - Схема структурна бізнес-процесу надання відповідей на питання користувачів

Послідовний опис процесів для відправлення відповідей на питання користувачів:

- якщо консультант авторизований у системі та має статус “активний”, тоді він має можливість переглядати питання користувачів, що стосуються його тематики;
- консультант переходить на вкладку “Dashboard”;
- на робочому столі у формі плиток будуть відображені усі питання користувачів з заголовком та тілом питання, що стосуються тематики консультанта;
- консультант має можливість натиснути на будь яке питання та надати свою відповідь у діалоговому вікні, що висвітиться у нього на робочому столі;

- після того, як консультант надрукує свою відповідь на питання він натискає кнопку “надіслати”;
- сервер приймає текст відповіді консультанта та іd питання, для того щоб зберегти його у базу;
- якщо сервер зберігає в базу питання, тоді питання, на яке відповів консультант пропадає з робочого стола та переміщається в вкладку “Архів”, де зберігаються усі питання консультанта;
- також консультант має можливість натиснути кнопку “Не цікаво”, що означає, що він не має бажання або не знає відповіді на питання, тоді питання автоматично пропаде з робочого стола.

Послідовний опис роботи консультанта з профілем:

- консультанту демонструється його профіль та перші дані, що ввів при реєстрації;
- консультант має можливість ввести чи змінити будь які поля продемонстровані в профілі;
- після того як консультант ввів всі дані та загрузив своє фото профілю, він має можливість натиснути кнопку “Зберегти”;
- у разі успішного збереження (якщо всі дані збереглися у БД) у консультанта з'явиться повідомлення про те, що його дані будуть автоматично змінені, коли модератор перевірить їх. Тобто дані консультанта не можливо змінити без верифікації модератором системи. У разі неуспішного збереження даних консультанту буде показано відповідне повідомлення у діалоговому вікні;
- коли модератор у адміністративній панелі верифікує змінні, тоді сервер перепише дані профілю консультанта.

Схема опису роботи консультанта з профілем зображена на рисунку 2.9

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

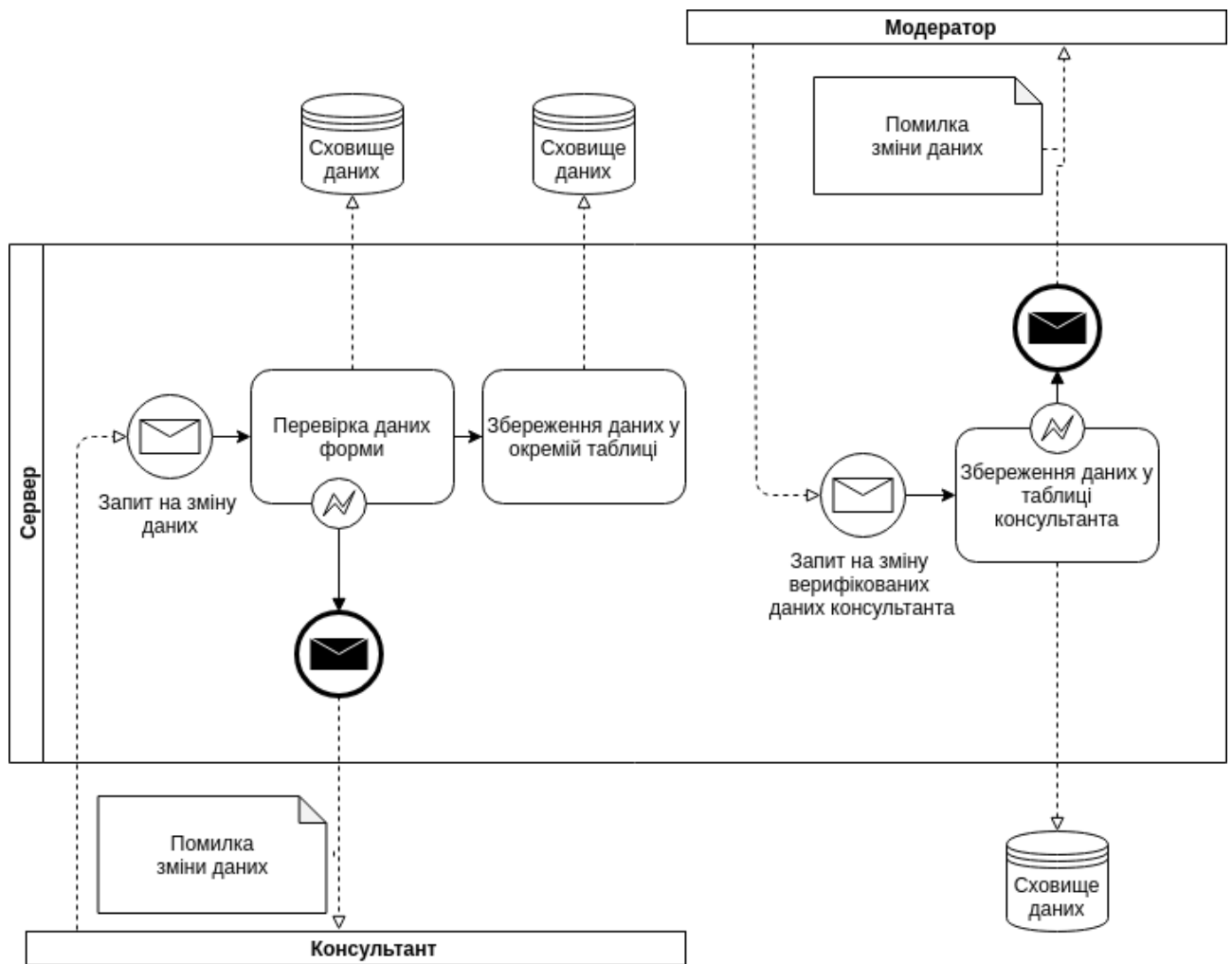


Рисунок 2.9 - Схема структурна бізнес-процесу роботи консультанта з профілем

Послідовний опис роботи консультанта з оповіщеннями:

- консультант натискає на кнопку налаштувань та обирає вкладку оповіщення;
- консультант має можливість пересунути повзунок налаштувань щодо оповіщень системних та оповіщень нових питань на його тематику;
- якщо повзунок пересунутий вправо, тоді функція активована, якщо повзунок пересунутий вліво та не підсвічений синім кольором, тоді функція вимкнена.

Схема опису роботи консультанта з оповіщенням зображена на рисунку 2.10

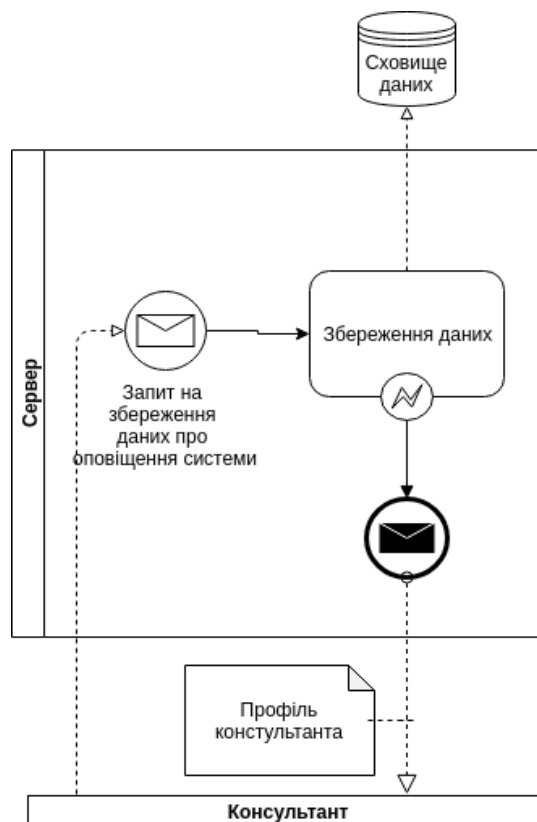


Рисунок 2.10 - Схема структурна бізнес-процесу роботи консультанта з оповіщенням

Послідовний опис роботи консультанта з підключенням карти:

- консультант натискає на кнопку налаштувань та обирає вкладку картки;
- консультант має можливість ввести дані картки у поле та натиснути “Зберегти”;
- якщо усі дані валідні та система wayforpay приймає дані картки, тоді дані картки зберігається у БД, а користувачу висвітлюється його дані, якщо виникає помилка або картка не валідна, тоді консультанту висвітлюється повідомлення у діалоговому вікні.

Схема опису роботи консультанта з підключенням карти зображена на рисунку 2.11

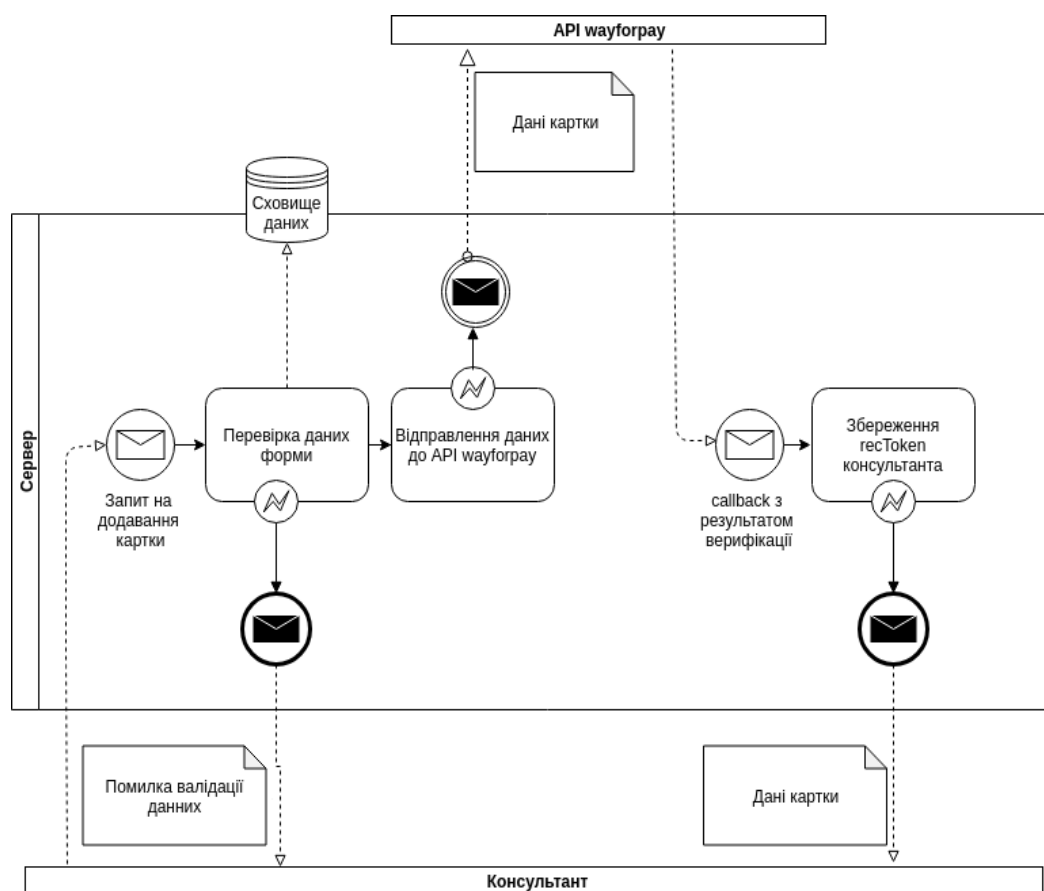


Рисунок 2.11 - Схема структурна бізнес-процесу роботи консультанта з підключенням карти

Додатковим сценарієм є процес зміни пароля.

Послідовний опис процесу зміни пароля:

- для зміни пароля консультант на вкладці авторизації натискає кнопку “забув пароль”;
- консультант вводить свою електронну пошту та натискає кнопку “Відправити посилання з відновленням пароля”;
- консультанту на пошту приходить повідомлення з посиланням на форму для зміни пароля та унікальним токеном користувача;
- після того як консультант клікне на посилання, він перенаправиться на форму зміни пароля;
- в формі консультант повинен ввести свою пошту, новий пароль та підтвердження нового пароля;

- після того, як консультант натисне кнопку “Відправити, на сервер вправляються такі дані, як новий пароль, підтвердження пароля, електронна пошта та токен;

- якщо всі дані, що сприймає сервер, вірні, та такий юзер у системі існує, тоді консультанту замінюється пароль на новий.

2.1.3. Адміністративна панель

Загальні задачі адміністративної панелі – це керування юзерами онлайн - платформи, а саме: модерація і валідація консультантів, створення нового користувача. Адміністративна панель має дві ролі для входу: адміністратор та модератор. Різниця їх ролей у тому, що адміністратор має можливість створювати нових користувачів, а модератор тільки валідувати і керувати зареєстрованими. Адміністратор і модератор також мають можливість ввійти у систему, але через інший роут та інтерфейс.

Послідовний опис процесів для авторизації адміністратора або модератора:

- адміністратор або модератор заповнює дані авторизаційної форми та відправляє на сервер;

- сервер валідує дані, отримані з фронтенду, перевіряє чи існує такий юзер у системі та перевіряє співпадіння пароля з електронною поштою;

- якщо електронна пошта та пароль співпадають, тоді адміністратор або модератор отримує токен, авторизується у системі та автоматично направляється в адміністративну панель.

Схема опису процесів для авторизації адміністратора або модератора зображена на рисунку 2.12

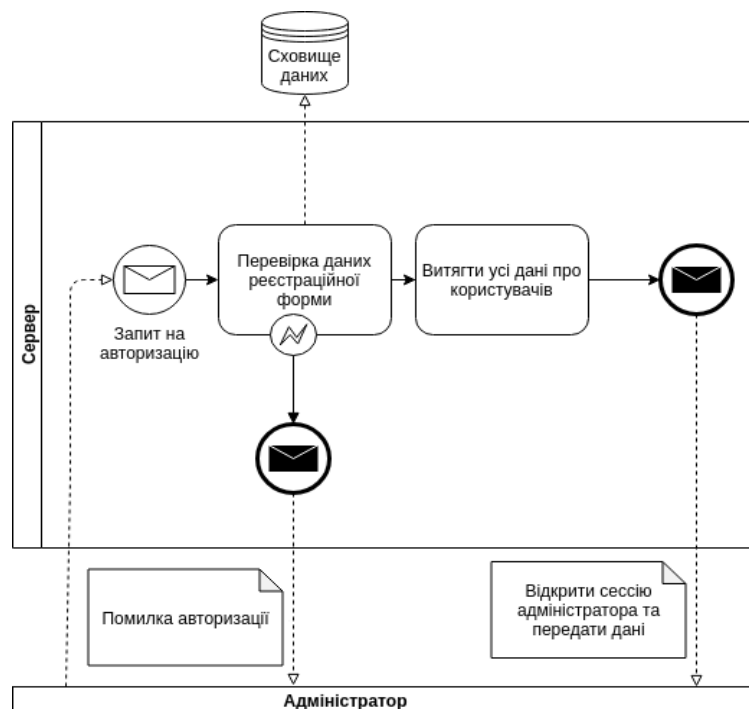


Рисунок 2.12 - Схема структурна бізнес-процесу авторизації адміністратора або модератора

Послідовний опис процесів для створення нового юзера адміністратором:

- адміністратор має бути авторизований у адміністративній панелі;
- адміністратор клікає на вкладку “Створити нового юзера”;
- на робочу столі з’являється форма для створення юзера;
- адміністратор заповнює обов’язкові поля та обирає роль нового юзера (модератор, клієнт, консультант);
- якщо всі дані вірні, та в системі не існує юзера з ідентичними даними, тоді сервер валідує нового юзера та записує його в БД, адміністратор отримає повідомлення про успішне створення та через 2 секунди перенаправить на панель з усіма юзерами сервісу;
- якщо дані не валідні, тоді адміністратор отримає повідомлення з описом помилки.

Схема опису процесів для створення нового юзера адміністратором зображена на рисунку 2.13

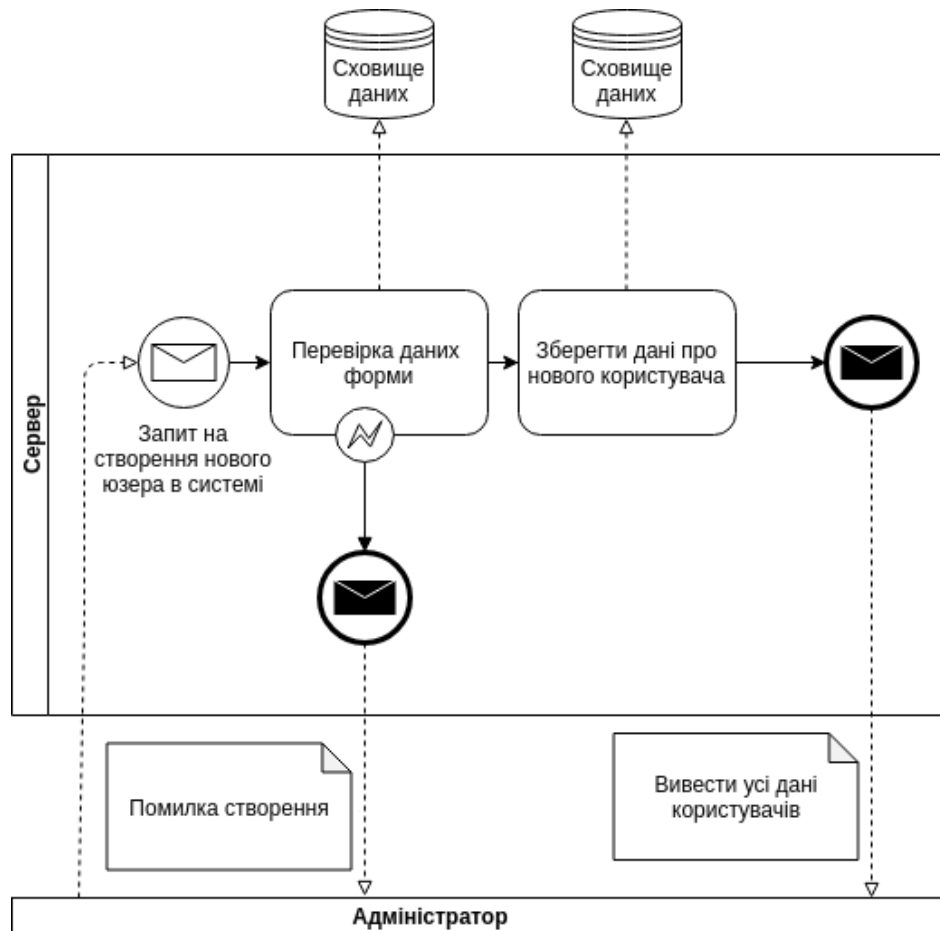


Рисунок 2.13 - Схема структурна бізнес-процесу створення нового юзера адміністратором

Послідовний опис процесів валідації новозареєстрованих консультантів модератором або адміністратором:

- адміністратор або модератор має бути авторизований у адміністративній панелі;
- адміністратор або модератор клікає на вкладку “Новозареєстровані консультанти”;
- на робочу столі з'являється список консультантів, що очікують на підтвердження;
- адміністратор або модератор переглядають інформацію про консультанта та завантажують документи, що прикріпив консультант при реєстрації;

- якщо документи та інформація консультанта задовільні, тоді модератор або консультант підтверджують статус консультанта як “Активний”;

- коли адміністратор або модератор натискають на кнопку “Підтвердити”, сервер змінює статус консультанта та змінює його роль з зареєстрованого на консультанта;

- якщо дані консультанта не є задовільні, тоді модератор або адміністратор переводять консультанта у заблоковані.

Послідовний опис процесів для валдіації консультантів, що бажають змінити інформацію про себе або діяльність модератором або адміністратором:

- адміністратор або модератор має бути авторизований у адміністративній панелі;

- адміністратор або модератор клікає на вкладку “Консультанти на модерації”;

- на робочому столі з’являється список консультантів, що вирішили змінити інформацію про себе;

- модератор або адміністратор можуть переглянути стару інформацію про консультанта та нову, котру він ввів про себе, та нові документи, що прикріпив консультант;

- якщо нова інформація про консультанта є задовільна, тоді модератор або адміністратор підтверджують дані, та сервер переписує дані профілю консультанта;

- якщо дані не є задовільні, модератор або адміністратор видаляють нову інформацію про консультанта та консультант залишається зі старими даними.

2.2 Архітектура програмного забезпечення

Онлайн-платформа містить наступні компоненти:

- ВЕБ-сервіс для консультантів;
- REST API я мобільних платформ;
- мобільний додаток для платформи iOS та Android для клієнтів;

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

- адміністративна панель;
- серверне застосування;
- сторонні сервіси, які взаємодіють з онлайн-платформою через REST або SOAP API.

Схема структурна розгортання з усіма компонентами платформи зображена на рисунку 2.14

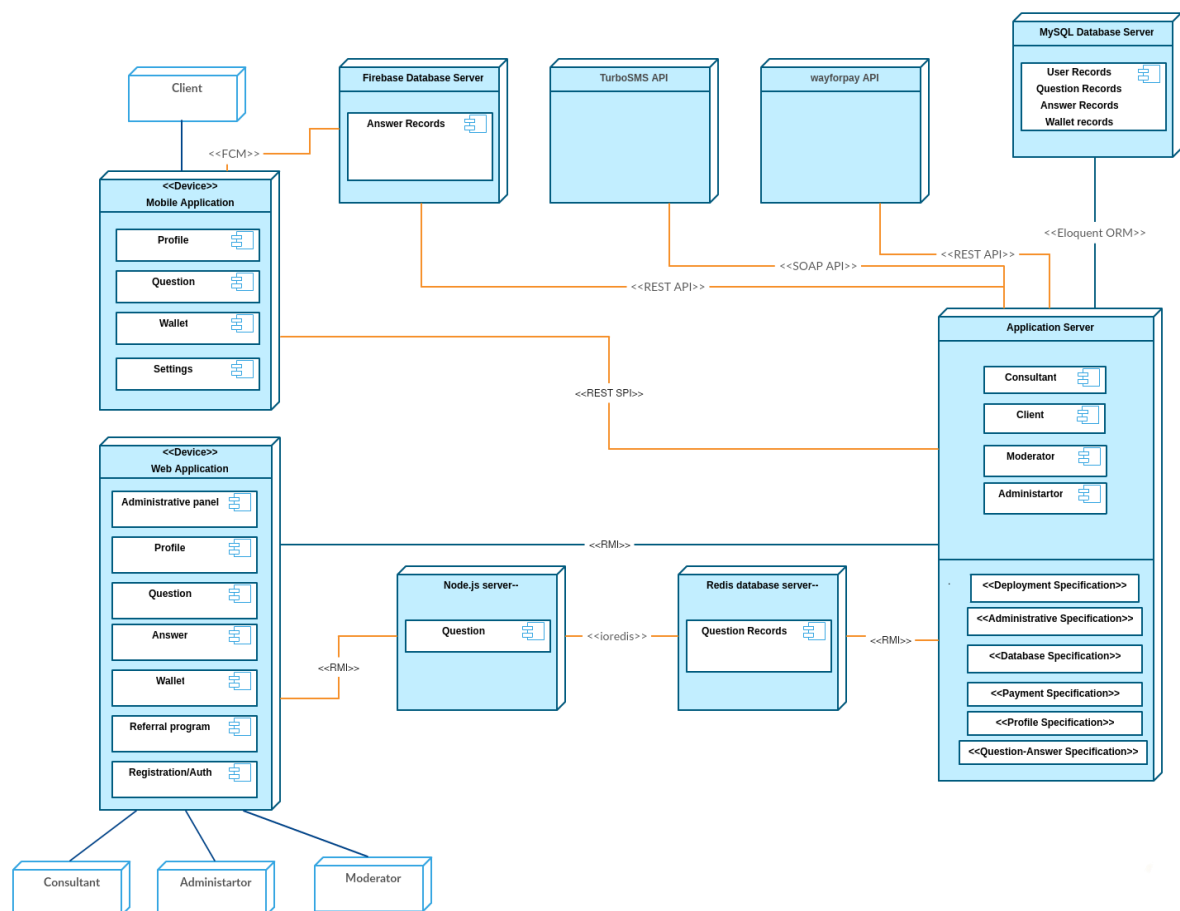


Рисунок 2.14 – Схема структурна розгортання платформи

2.3 Конструювання програмного забезпечення

Для вирішення поставленого завдання було обрано клієнт-серверну модель системи, яку продемонстровано на рисунку 2.15

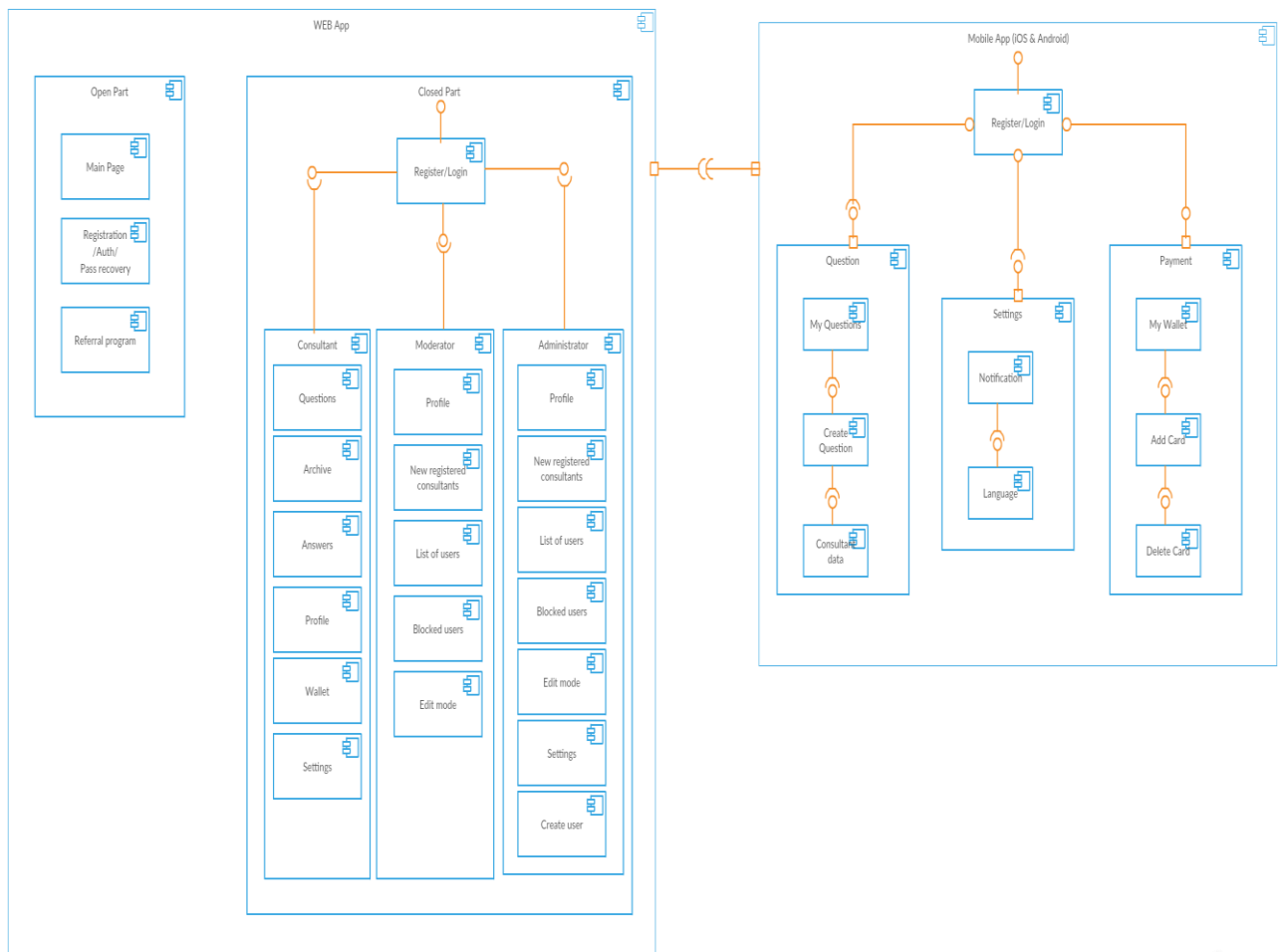


Рисунок 2.15 – Схема структурна компонентів мобільного додатку та веб-сервісу

Взаємодія між компонентами системи здійснюється за допомогою мережевого протоколу JSON над TLS над TCP.

Для взаємодії з клієнтським додатком використовують наступні інтерфейси:

Auth – інтерфейс для процесів авторизації, аутентифікації, реєстрації та підтвердження особистих даних, відновлення та зміни пароля;

Config – інтерфейс для отримання конфігурації додатку;

Question — інтерфейс для отримання та збереження питань;

Answer — інтерфейс для отримання та збереження відповідей;

Wallet — інтерфейс для збереження даних кредитної картки та інтеграції з API wayforpay;

Profiles — інтерфейс для відображення та збереження даних профілю;
Admin – інтерфейс для взаємодії адміністратора із адміністративною панеллю.

Детальна інформація про методи інтерфейсів наведена у таблицях 2.1- 2.6

Таблиця 2.1 - Методи інтерфейсу Auth

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
authenticate	credentials	token	Аутентифікувати користувача за вказаними відповідями на питання аутентифікації
register	email, password	–	Зареєструвати особисті дані користувача
confirmEmail	token	–	Підтвердити email
login	credentials, password	token	Увійти у застосунок за вказаними відповідями на питання авторизації і паролем
restorePassword	email	–	Відновити пароль за вказаним email

Таблиця 2.2 – Методи інтерфейсу Question

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
index	token	Список питань у форматі JSON	Показати усі питання користувачв
show	Token, question_id	Питання з усіма відповідями консультантів та профілі консультантів	Показати відповіді на конкретне питання з короткою інформацією про консультантів
store	Token, topic_id, quesion, title	Інформація з питання	Зберегти питання у БД для консультантів
ShowQuestionforUser	token	Список питань по тематики консультанта у інтерфесі веб сервісу	Відобразити всі питання для консультанта

Таблиця 2.3 - Методи інтерфейсу Answer

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
getAnswers	Token, question_id	Список усіх питань по конкретному питанню	Отримати список відповідей на питання клієнта

Продовження таблиці 2.3

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
chooseBest Answer	Token, question_id, answer_id	Питання у форматі JSON	Обрати найкращу відповідь на питання клієнта
showArchiv eforUser	token	Список відповідей консультанта у інтерфесі веб сервіса	Відобразити консультанту список всіх його відповідей на питання
store	Token, question_id, answer	Перенаправлен ня на список усіх питань	Надати відповідь клієнту та відправити Push- повідомлення
declineAnsw er	Token, question_id	-	Додати питання у стопліст та прибрати з дешборда для консультанта

Таблиця 2.4 – Методи інтерфейсу Wallet

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
addCard	Token, cardnumber, expMonth, expYear, cvv, cardHolder	Повідомлення від API wayforpay у форматі JSON	Додати картку до системи
showCardCred	Token	4 останні цифри картки, місяць та рік завершення картки у форматі JSON	Показати користувачу безпечні дані кредитної картки
RemoveCard	Token	-	Видалити картку користувача
TransactFromCardToCard	Token, currency, recToken, amount, order, way_for_pay_key, way_for_pay_domain	Повідомлення від API wayforpay у форматі JSON	Перевести гроші з картки на картку

Продовження таблиці 2.4

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
TransactFromAccountToCard	Token, currency, recToken, amount, order, way_for_pay_ key, way_for_pay_ domain	Повідомлення від API wayforpay у форматі JSON	Перевести гроші з корпоративного рахунку на картку

Таблиця 2.5 – Методи інтерфейсу Profile

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
showProfile	token	Профіль користувача у форматі JSON	Відобразити користувачу його профіль
showReferralCode	Token	Реферальний код користувача	Відобразити користувачу його реферальний код
setNotifications	Token, notification_client, notification_system	Профіль користувача у форматі JSON	Зберегти налаштування оповіщень користувача

Продовження таблиці 2.5

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
setProfile	Token, first_name, last_name, middle_name, email, city, password	Профіль користувача у форматі JSON	Змінити інформацію профіля користувача
uploadImage	Token, file(„avatar“)	Профіль користувача у форматі JSON	Загрузити аватар користувача
showConsultantProfile	Token, user_id	Профіль консультанта у форматі JSON	Відобразити клієнту профіль користувача, дані котрого він відкрив
changeLanguage	Token, language	Профіль користувача у форматі JSON	Змінити мову мобільного додатку клієнта та запам'ятати її
setDeviceToken	Token, device_token	-	Зберегти унікальний код Firebase клієнта для Push-повідомлень

Продовження таблиці 2.5

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
downloadCertificate	Token, certificate	Загрузка документа консультанта	Можливість завантажити документ консультанта. Можливість надається як консультанту так і модераторам

Таблиця 2.6 – Методи інтерфейсу Admin

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
ChangeOnModerator	Token, type, city, workingTime, callingTime, email, phone	Адміністратора перенаправляє на список консультантів	Верифікувати новозареєстрованого консультанта та надати йому статус “активний”
index	token	Список усіх юзерів системи у інтерфейсі	Отримати весь список юзерів платформи

Продовження таблиці 2.6

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
ShowClients	token	Список усіх клієнтів у інтерфейсі	Відобразити користувачів мобільного додатку
showConsultants	token	Список усіх консультантів у інтерфейсі	Відобразити верифікованих консультантів
showConsultants onModerating	token	Список усіх консультантів, що вирішили модифікувати свій профіль у інтерфейсі	Відобразити список консультантів, що вирішили змінити інформацію свого профілю
showConsultants Unverified	token	Список усіх неверифікованих консультантів у інтерфейсі	Відобразити консультантів, що не пройшли верифікацію модератором

Продовження таблиці 2.6

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
store	Token, first_name, last_name, email,phone, password, password_confir mation, role	Список усіх юзерів у інтерфейсі	Зберегти новго юзера у системі
acceptChang es	token, user_id	Список усіх верифікованих консультантів у інтерфейсі	Прийняти зміни профіля консультанта
declineChan ges	Token, user_id	Список усіх неверифікованих консультантів у інтерфейсі	Відхилити зміни профіля консультанта
show	Token, user_id	Інформація одного юзера системи у інтерфейсі веб сервіса	Відобразити інформацію про конкретного юзера системи
downloadDo cument	Token, document_id	Завантаження документа	Завантажити документ консультанта

Продовження таблиці 2.6

Назва метода	Аргументи	Значення, яке повертається	Призначення методу
search	Token, search_item	Список юзерів, що відповідають пошуку у інтерфейсі веб сервіса	Зробити пошук серед усіх зареєстрованих юзерів системи по імені, фамілії чи електронній пошті

Структури і типи даних, які відправляються і отримуються у запитах, вказано в таблиці 2.7

Таблиця 2.7 – Структури і типи даних

Назва типу	Структура та / або тип даних	Опис даних
Question	{ title: 'string', user_id: 'int' topic_id: 'int', question: 'string', best_answer_chosen: 'boolean', created_at: 'timestamp' }	Інформація про питання. Містить назву питання, тематику питання, саме питання, id користувача, що створив питання, помітка, чи обрана була найкраща відповідь на питання, час створення питання

Продовження таблиці 2.7

Назва типу	Структура та / або тип даних	Опис даних
Answer	{ question_id: 'int', user_id: 'int' answer: 'string', best_answer: 'boolean', created_at: 'timestamp' }	Відповідь на питання. Містить id питання, на яке була надана ця відповідь, текст відповіді на питання, id користувача, що відповів на питання, помітка, чи обрана була відповідь як найкраща, час створення питання
Document	{ user_id: 'int' certificate: 'string', working_book: 'string', created_at: 'timestamp' }	Документ консультанта. Містить id консультанта, шлях до документа(сертифіката) на сервері, шлях до документа(трудова книга) на сервері
Wallet	{ user_id: 'int' rec_token: 'string', card_last_nums: 'string', card_expired_month: 'string', card_expired_year: 'string', card_type: 'string', created_at: 'timestamp' }	Картка користувача. Містить інформацію про картку, а саме id юзера, токен API wayforpay, останні 4 цифри картки, місяць завершення дії картки, рік завершення дії картки, тип картки, час прикріплення картки до системи

Продовження таблиці 2.7

Назва типу	Структура та / або тип даних	Опис даних
Config	Файли з конфігураціями	Конфігурація додатку. В кожному файлі містяться дані паролів, ключів до API, ключі сервісу, токени та налаштування

2.4 Аналіз безпеки даних

Для клієнт-серверної частини використовується мережевий протокол HTTPS.

HTTPS (англ. HyperText Transfer Protocol Secure) - розширення HTTP, яке підтримує захист даних при транспортуванні за допомогою шифрування інформації відповідно до стандартів SSL і TLS. Такий захист необхідний в комерційних ресурсах, де використовується конфіденційна інформація про персональні або платежів користувача. Різниця між HTTP і HTTPS ще в тому, що основний протокол використовує порт 80, розширений - порт 443. [8].

TLS – криптографічний протокол, що надає можливості безпечної передачі даних в Інтернет для навігації, отримання пошти, спілкування, обміну файлами, тощо [6]. Він додає додатковий шар шифрування і аутентифікації між HTTP та TCP.

TCP – мережевий протокол над протоколом IP, призначений для управління передачею даних у комп'ютерних мережах [5].

Для збереження налаштувань у серверній частині застосунку використовуються config та .env-файли. Це пари ключів-значень, які доступні компонентам застосунку, які мають відповідний контекст. Вони зберігаються як файл у файловій системі.

Для збереження паролів використовуються хешування паролів за допомогою алгоритму bcrypt. bcrypt був вибраний так як він є одним з алгоритмів, рекомендованих OWASP, для хешування паролів [4].

bcrypt – адаптивна криптографічна функція формування ключа, що використовується для безпечного зберігання паролів. Розробники: Нілс Провос і David Mazières. Функція заснована на шифрі Blowfish, вперше представлена на USENIX у 1999 році. Для захисту від атак за допомогою райдужних таблиць bcrypt використовує сіль (salt); крім того, функція є адаптивною, час її роботи легко налаштовується і її можна сповільнити, щоб ускладнити атаки перебором [2]

2.5 Висновки по розділу

У даному розділі було проаналізовано бізнес-процеси інформаційної системи онлайн-платформи, змодельовано та проілюстровано з використанням діаграм.

Було розроблено принципову схему клієнт-серверної моделі, описано і розроблено API для мобільних платформ та побудовано загальну архітектуру платформи з інтеграцією сторонніх сервісів за допомогою REST та SOAP API.

Було зконструйовано принцип взаємодії між компонентами системи, типи та структури даних для успішної роботи платформи.

Було обрано рішення для забезпечення безпеки даних під час зберігання та передачі між компонентами системи

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Ціллю даного тест плану являється опис процесу тестування онлайн - платформи для надання юридичних послуг.

Мають досягатися наступні цілі:

- а) описати дії, необхідні для підготовки до тестування;
- б) визначити джерела інформації необхідні для створення плану тестування;
- в) визначити інструменти та середовище тестування.

Для закінчення бакалаврського курсу була визначена тема дипломного проекту: «Онлайн - платформа для надання юридичних послуг». Результатом виконання бакалаврської роботи є готовий програмний продукт (ПЗ). Був створений план розробки ПЗ, технічні вимоги, ескізний проект та деяка частина функціоналу ПЗ. Виникла необхідність тестування ПЗ.

Тестовий план створений задля тестування ПЗ «Онлайн - платформа для надання юридичних послуг» в цілому та її окремих модулів, контролю та планування тестування, передбачення результатів тестування. Програмний продукт передбачає обробку зображень великої роздільної здатності та побудований на клієнт-серверній архітектурі.

Сервер має HTTP API, яке базується на JSON та веб - сервіс, що базується на використанні сесій.

Усі елементи, з яких складається «Онлайн - платформа для надання юридичних послуг» мають бути протестовані. Перерахування основних складових елементів ПЗ знаходяться у таблиці 3.1.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

Таблиця 3.1 - Основні складові елементів ПЗ

Елемент	Опис у документі	Вимоги до платформи
БД «Платформи »	DC001V1-01	Розташовано на сервері
Серверна частина додатку, що здійснює безпосередню роботу з БД	DC001V1-02	Розташовано на сервері
Інтерфейс сайту	DC001V1-03	Розташовано на персональному комп'ютері
Функціонал сайту	DC001V1-04	Розташовано на персональному комп'ютері
Алгоритми створення питань	DC001V1-05	Розташовано на сервері
Алгоритми створення відповідей	DC001V1-06	Розташовано на сервері
Система надсилення Push-повідомлень	DC001V1-07	Розташовано на сервері
Інтеграція з API wayforpay та SMSturbo	DC001V1-08	Розташовано на сервері

3.2 Опис процесів тестування

Димне тестування буде запускатися кожного разу перед деплоєм нової версії на Debug конфігурації, та після деплоя на Release. У випадку Release, усі тести будуть проводитись в окремій базі даних.

3.2.1 Тестування конвертації та обміну даними

- Реєстрація - створення нового користувача, перевірка функції «Забув пароль». Система має вислати посилання на пошту з можливістю створити новий пароль.
- Логінізація – вхід в систему. Успішний вхід в систему при коректно введених даних.
- Створення питання – створення нового питання. Система має додавати усі дані в БД та відобразити новостворене питання у дашборді консультанта.
- Створення відповіді – можливість зберегти нову відповідь на питання консультантом. Система має додавати усі дані в БД та відобразити новостворену відповідь у дашборді користувача.
- Система надсилання Push-повідомлень – Користувачу у мобільний додаток має приходити Push-повідомлення через БД Firebase при кожній новій відповіді на його питання.
- Інтеграція з API wayforpay та SMSturbo - Користувачу має приходити код підтвердження при реєстрації у мобільному застосунку. Система wayforpay має бути інтегрована з серверним додатком, та при валідно введених даних кредитної картки має списуватись сума вартості послуги.

3.2.2 Тестування зручності інтерфейсу

Інтерфейс має бути зручним для користувача, задля чого проводиться тестування інтерфейсу різними користувачами, виділення «гарячих зон» та найдовших шляхів.

3.2.3 Тестування інтерфейсу користувача

Виконання тестів при цьому виробляється системою, що емулює дії користувача для виявлення помилок у функціональності, необроблених виключень при взаємодії з інтерфейсом, втраті або переключення даних, а також в ручному режимі для перевірки відсутності певних елементів інтерфейсів.

3.2.4 Тестування функціоналу сайту та додатку (системне тестування)

Уся обробка даних може бути покрита автоматизованими тестами, оскільки обробка даних буде проводитись на стороні серверу та взаємодія компонентів додатку і передача даних буде відбуватись через REST API. Оскільки ми знаємо які дані повинна повертати кожна функція і метод, ми можемо автоматизувати дані тести.

3.2.5 Тестування навантаження

Система має задовольнити вимогам щодо навантаження, які вказані у Технічному завданні.

3.2.6 Тестування безпеки

Коректна робота з паролями, логінами – валідація усіх даних

Захищений доступ до бази даних – захист перед SQL- ін'єкціями.

3.2.7 Тестування стабільності

Система має працювати без збоїв протягом місяця.

Система має виконувати усі вимоги, що вказані у Технічному завданні та Документації ПЗ:

- у системі не має бути знайдено нових помилок;
- відсутність критичних та мажорних помилок в цілому;
- загальна кількість пройдених тестів має бути рівною 100%.

Критерій призупинення тестування:

- наявність помилки, що блокує роботи всієї програми;
- наявність помилки, що блокує роботи всієї програми;
- може бути призупинене тестування деякої частини ПЗ при знаходженні у ній критичної помилки, що блокує подальше тестування (помилки ініціалізації і т.д.).

Усі частини можуть бути протестовані окремо, отже повна зупинка процесу тестування малоімовірна.

Критерій відновлення тестування:

- виправлення критичної помилки.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

3.3 Опис контрольного прикладу

Обладнання:

- частина ПЗ, що розташована на сервері має бути протестована на сервері. Вимоги до сервера вказані у Технічному завданні;
- частина ПЗ, що розташована на клієнтській машині має бути протестована на клієнтській машині. Вимоги до клієнтської машини вказані у Технічному завданні.

Програмне забезпечення:

- повинно бути встановлене програмне забезпечення вказане у документації.

Інструменти:

- SOAPUI для перевірки повідомлень;
- UserTesting для тестування UI та UX;
- SQL Server Management Studio для тестування БД;
- інші функції тестуються на тому ж інструменті, на якому були розроблені за допомогою xTest.

В процесі тестування була перевірена уся основна функціональність системи. Послідовно були перевірені всі варіанти використання, результати представлені у відповідних таблицях:

- авторизація користувача (таблиця 3.2);
- створення питання клієнтом (таблиця 3.3);
- надання відповіді консультантом (таблиця 3.4);
- отримання Push-повідомлення клієнтом (таблиця 3.5);
- зняття грошей з рахунку користувача (таблиця 3.6);
- валідація консультанта у адміністративній панелі (таблиця 3.7).

Таблиця 3.2 – Перевірка можливості авторизації користувача

Мета тесту	Перевірка можливості авторизації користувача
Початковий стан	
Вхідні дані	Номер телефону.
Схема проведення тесту	Відкрити мобільний додаток. Натиснути кнопку “Авторизуватись”. Ввести номер телефону у поле. Натиснути кнопку “Авторизуватись”. Отримати смс-код підтвердження. Ввести код-підтвердження у відповідне поле. Натиснути кнопку “Продовжити”.
Очікуваний результат	Клієнта перенаправлено до головної сторінки, де можна задати питання. Усі дані клієнта передано до додатку та валідно відображаються у профілі користувача.
Стан програмного продукту після проведення випробувань	Клієнта перенаправлено до головної сторінки, де можна задати питання. Усі дані клієнта передано до додатку та валідно відображаються у профілі користувача.

Таблиця 3.3 – Перевірка можливості створення питання клієнтом

Мета тесту	Перевірка можливості створення питання клієнтом
Початковий стан	Відкритий мобільний додаток; користувач авторизований як клієнт.
Вхідні дані	Заголовок питання, тема, саме питання.
Схема проведення тесту	Натиснути кнопку “Створити питання”. Ввести дані форми, а саме, тема питання, заголовок та саме питання. Натиснути кнопку “Створити питання”

Продовження таблиці 3.3

Мета тесту	Перевірка можливості створення питання клієнтом
Очікуваний результат	Нове питання відображається у формі плашки у дашборді користувача. Питання активне та є доступним у консультантів.
Стан програмного продукту після проведення випробувань	Нове питання відображається у формі плашки у дашборді користувача. Питання активне та є доступним у консультантів.

Таблиця 3.4 – Перевірка можливості надання відповіді консультантом

Мета тесту	Перевірка можливості надання відповіді консультантом
Початковий стан	Відкрита веб-сторінка консультанта; авторизований у системі консультант; відкрита вкладка усіх питань користувачів.
Вхідні дані	Текстова відповідь на запитання.
Схема проведення тесту	Обрати бажане питання. Натиснути на кнопку “Відповісти”. Ввести текст у діалогове вікно. Натиснути кнопку “Відправити відповідь”.
Очікуваний результат	Консультанта вертає на дашборд з усіма відповідями, що залишились. Питання пропало з вкладки усіх питань та збереглось у вкладці “Архів”. Клієнт може переглянути дану відповідь на питання у мобільному додатку.
Стан програмного продукту після проведення випробувань	Консультанта вертає на дашборд з усіма відповідями, що залишились. Питання пропало з вкладки усіх питань та збереглось у вкладці “Архів”. Клієнт може переглянути дану відповідь на питання у мобільному додатку.

Таблиця 3.5– Перевірка отримання Push-повідомлення клієнтом

Мета тесту	Перевірка отримання Push-повідомлення клієнтом
Початковий стан	
Вхідні дані	
Схема проведення тесту	Консультант обирає бажане питання. Натиснути на кнопку “Відповісти”. Ввести текст у діалогове вікно. Натиснути кнопку “Відправити відповідь”.
Очікуваний результат	Користувач отримає повідомлення на екрані телефона з текстом “Отримана нова відповідь на питання”
Стан програмного продукту після випробувань	Користувач отримає повідомлення на екрані телефона з текстом “Отримана нова відповідь на питання”

Таблиця 3.6 – Перевірка зняття грошей з рахунку користувача

Мета тесту	Перевірка зняття грошей з рахунку користувача
Початковий стан	Відкритий мобільний додаток; Авторизований користувач як клієнт; Створене питання користувачем; Список відповідей на задане питання.
Вхідні дані	
Схема проведення тесту	Користувач обирає відповідь, що йому сподобалась. Натискає на відповідь. Натиснути кнопку “Відкрити дані консультанта”.
Очікуваний результат	Система відображає дані консультанта. Система за допомогою API wayforpay списує 1 грн з рахунку користувача.

Продовження таблиці 3.6

Мета тесту	Перевірка зняття грошей з рахунку користувача
Стан програмного продукту після проведення випробувань	Система відображає дані консультанта. Система за допомогою API wayforpay списує 1 грн з рахунку користувача.

Таблиця 3.7 – Перевірка можливості валідувати консультанта у адміністративній панелі

Мета тесту	Перевірка можливості валідувати консультанта у адміністративній панелі
Початковий стан	Відкрита вкладка усіх консультантів у адміністративній панелі; Користувач авторизований як адміністратор або модератор.
Вхідні дані	
Схема проведення тесту	Обрати новозареєстрованого консультанта. Натиснути на кнопку “Переглянути дані”. Натиснути кнопку “Підтвердити”.
Очікуваний результат	Система змінює статус консультанта як “Активний”. Дані консультанта пропадають зі списку новозареєстрованих консультантів, та з’являються у списку підтверджених. Статус консультанта у профілі відображається як “Активний статус”. Консультант може відповідати на питання клієнтів.

Продовження таблиці 3.7

Мета тесту	Перевірка можливості валідувати консультанта у адміністративній панелі
Стан програмного продукту після проведення випробувань	Система змінює статус консультанта як “Активний”. Дані консультанта пропадають зі списку новозареєстрованих консультантів, та з'являються у списку підтверджених. Статус консультанта у профілі відображається як “Активний статус”. Консультант може відповідати на питання клієнтів.

3.4 Висновки по розділу

У даному розділі було проаналізовано якість програмного забезпечення, а саме описано процес тестування онлайн-платформи. Були описані цілі для успішного тестування платформи, межі застосування та основні складові елементи, що тестуються.

Було надано опис процесів тестування, та вимоги, що має виконувати система та результати тестування цих вимог.

Був наданий опис контрольного прикладу, а саме була перевірена уся функціональність підсистеми. Послідовно були перевірені всі основні варіанти використання, результати були представлені у відповідних таблицях.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Для розгортання серверної частини програмного забезпечення, необхідною є наявність Node.js, менеджера пакетів NPM, PHP7.2 або вище, composer, Laravel5.7 або вище, Redis, MySQL.

Для встановлення платформи потрібен сервер Apache або nginx. На сервер в першу чергу необхідно встановити PHP7.2 та Node.js. Далі встановити фреймворк Laravel та всі необхідні модулі. Після успішного встановлення необхідно перенести усі файли Node.js Laravel у відповідні папки на сервері (Laravel у папку з назвою домену, а файли Node.js у сусідню для зручності). Після переносу усіх файлів необхідно налаштувати усі доступи до Redis, MySQL та ін. Надалі виконати міграції БД та при потребі налаштувати файл .htaccess. Після успішного виконання усіх вище описаних операцій Веб-сервіс буде готовий до роботи та мобільні додатки зможуть використовувати REST API платформи для роботи.

4.2 Робота з програмним забезпеченням

Детальну інструкцію роботи із клієнтською частиною програмного забезпечення наведено у документі “Керівництво користувача”.

4.3 Висновки по розділу

У даному розділі детально описаний процес розгортання платформи, а саме клієнтської і серверної частини, перераховані усі системні вимоги для успішного розгортання проекту, а також описано засоби роботи з програмним забезпеченням.

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		86

Також було надано документацію по роботі з платформою, а саме керівництво користувача, що включає в себе інструкцію роботи адміністратора, клієнта, модератора та адміністратора платформи.

					КПІ.ІП-5121.045440.01.81	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході дипломного проекту було проведено аналіз проблематики онлайн - пошуку консультанта у сфері юриспруденції, а також огляд існуючих програмних рішень .

Проведено аналіз вимог до програмного забезпечення, на основі яких були спроектовані наступні компоненти онлайн-платформи: серверне застосування, веб-сервіс, адміністративна панель, API для мобільних платформ з операційною системою Android та iOS. Також реалізована інтеграція сторонніх сервісів для підтримки дієздатності платформи.

Розроблена платформа має ряд переваг, які не було реалізовано в проаналізованих існуючих аналогах, а саме:

- можливість валідації та верифікації консультантів через адміністративну панель;
- підтримка API для мобільних платформ з можливістю отримувати Push-повідомлення.

Також проведено аналіз якості розробленого програмного забезпечення, виконано його тестування.

Розроблена онлайн-платформа має практичне застосування і вже працює в онлайн - режимі, та розгорнута на VPS сервері, та може бути доступна за посиланням <https://consultant.net.ua>

					КП.ІП-5121.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		88

ПЕРЕЛІК ПОСИЛАНЬ

1) <http://loyer.com.ua/uk/yuridichni-sajti-portali-ta-resursi-100-korisnih-posilan-dlya-yuristiv/>

2) Bcrypt [Електронний ресурс]: (Стаття) / Wikipedia. – Електрон. дан. (1 файл). – 2018. – Режим доступу: <https://uk.wikipedia.org/wiki/Bcrypt>. – Назва з екрана.

3) <https://zkg.ua/yurydychni-posluhy-praktyky/yurydychni-posluhy-dlya-tvorchih-tehnolohichnyh-industrij/>

4) Password storage cheat sheet [Електронний ресурс]: (Стаття) / Owasp. – Електрон. дан. (1 файл). – 2018. – Режим доступу: https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet#Use_a_cryptographically_strong_credential-specific_salt. – Назва з екрана.

5) TCP [Електронний ресурс]: (Стаття) / Wikipedia. – Електрон. дан. (1 файл). – 2018. – Режим доступу: https://uk.wikipedia.org/wiki/Transmission_Control_Protocol. – Назва з екрана.

6) TLS [Електронний ресурс]: (Стаття) / Wikipedia. – Електрон. дан. (1 файл). – 2018. – Режим доступу: https://uk.wikipedia.org/wiki/Transport_Layer_Security. – Назва з екрана.

7) <https://home.kpmg/ua/uk/home/services.html>

8) <https://cityhost.ua/blog/chto-takoe-https.html>

9) <https://laravel.com/docs/5.8>

10) <https://laracasts.com>

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

ОНЛАЙН-ПЛАТФОРМА ДЛЯ НАДАННЯ ЮРИДИЧНИХ
ПОСЛУГ

Технічне завдання

КП.ІП-5121.045440.02.91

“ПОГОДЖЕНО”

Керівник проекту:

_____ І.П. Муха

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ О.В. Субцельний

Київ – 2019 року

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
4.1	Вимоги до функціональних характеристик.....	6
4.2	Вимоги до надійності	7
4.3	Умови експлуатації	7
4.4	Вимоги до складу і параметрів технічних засобів	7
4.5	Вимоги до інформаційної та програмної сумісності	8
4.6	Вимоги до маркування та пакування.....	8
4.7	Вимоги до транспортування та зберігання	8
4.8	Спеціальні вимоги.....	8
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	9
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	10
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	12

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки:

Галузь застосування:

Наведене технічне завдання поширюється на розробку програмного забезпечення «Онлайн-платформа для надання юридичних послуг», котра використовується для надання можливості консультації з юристом, адвокатом, бухгалтером в онлайн-режимі та призначена для зв'язку клієнта і юриста / адвоката / бухгалтера для надання допомоги клієнту при виникненні будь-яких питань, пов'язаних з юриспруденцією і бухгалтерією.

					КПІ.ІП-5121.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки «Онлайн-платформа для надання юридичних послуг» є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім.Ігоря Сікорського).

					КПІ.ІП-5121.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для надання можливості консультації з юристом, адвокатом, бухгалтером в онлайн -режимі.

Метою розробки є зв'язок клієнта і юриста / адвоката / бухгалтера для надання допомоги клієнту при виникненні будь-яких питань, пов'язаних з юриспруденцією і бухгалтерією.

					КПІ.ІП-5121.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик:

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1.1 для користувача:

- реєстрація та авторизація;
- можливість зміни мобільного телефону реєстрації;
- задати запитання;
- переглянути питання та відповіді;
- відкрити сторінку консультанта;
- прив'язати кредитну картку;
- переглянути та змінити персональні дані.

4.1.1.2 Для консультанта системи:

- реєстрація та авторизація;
- дати відповідь на запитання користувача;
- переглянути усі відповіді на питання;
- прив'язати кредитну картку;
- переглянути та змінити персональні дані.

4.1.1.3 Для адміністратора системи:

- авторизація;
- валідація, перевірка та погодження нового консультанта у системі;
- блокування консультанту з системи;
- створити нового юзера у системі.

4.1.2 Розробку виконати на платформі Laravel та у формі REST API.

4.1.3 Додаткові вимоги:

- використання сесійного режиму;
- зберігання важливих даних користувача на веб-сервісу в зашифрованому вигляді;
- підтримка усіх розширення екрану;
- підтримка усіма сучасними браузерями.

4.2 Вимоги до надійності:

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.2.3 Забезпечити цілісність інформації в базі даних.

4.3 Умови експлуатації.

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.4 Вимоги до складу і параметрів технічних засобів.

4.4.1 Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.

4.4.2 Мінімальна конфігурація технічних засобів:

Тип процесору Pentium.

Об'єм ОЗП 256 Мб

					КПІ.ІП-5121.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення веб-ресурсу повинно працювати під управлінням операційних систем сімейства WIN32 (Windows'XP, Windows NT і т.д.) або Unix, та API мобільного додатку на iOS, Android.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі: POST запиту у вигляді JSON формату.

4.5.3 Результати повинні бути представлені в наступному форматі: список представлений у вигляді об'єкта JSON формату.

4.6 Вимоги до маркування та пакування.

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання.

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги.

Згенерувати установчу версію програмного забезпечення.

					КПІ.ІП-5121.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему.

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка формату А4.

5.3.2 Технічне завдання.

5.3.3 Опис програми.

5.3.4 Програма та методика тестування.

5.3.5 Керівництво користувача.

5.4 Графічна частина повинна бути виконана на листі формату А3.

5.4.1 Схема структурна розгортання.

5.4.2 Схема структурна компонентів програмного забезпечення.

5.4.3 Схема бази даних.

5.4.4 Схема структурна класів програмного забезпечення.

5.4.5 Креслення вигляду екранних форм.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

	Назва етапу	Строк	Звітність
.	Вивчення літератури за тематикою проекту	01.04.19	
.	Розробка технічного завдання	15.04.19	Технічне завдання
.	Аналіз вимог та уточнення специфікацій	18.04.19	Специфікації програмного забезпечення
.	Проектування структури програмного забезпечення, проектування компонентів	28.04.19	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму ...)
.	Програмна реалізація програмного забезпечення	15.05.19	Тексти програмного забезпечення
.	Тестування програмного забезпечення	19.05.19	Тести, результати тестування
.	Розробка матеріалів текстової частини проекту	20.05.19	Пояснювальна записка.
.	Розробка матеріалів графічної частини проекту	22.05.19	Графічний матеріал проекту

КПІ.ІП-5121.045440.02.91

.	Оформлення технічної документації проекту	24.05.19	Технічна документація
---	---	----------	--------------------------

					КПІ.ІП-5121.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-5121.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

ОНЛАЙН – ПЛАТФОРМА ДЛЯ НАДАННЯ ЮРИДИЧНИХ ПОСЛУГ

Опис програми

КПІ.ІП-5121.045440.03.13

“ПОГОДЖЕНО”

Керівник проекту:

_____ І.П. Муха

Виконавець:

_____ О.В. Субцельний

Нормоконтроль:

_____ К.І. Ліщук

Київ – 2019 року

Тексти програмного коду**Онлайн – платформа для надання юридичних послуг**

(Найменування програми (документа))

DVD-R

(Вид носія даних)

43 арк, 276,9 Мб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019

					КПІ.ІП-5121.045440.03.13	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Main Payment Logic Model:

WalletLogicModel

```
<?php
```

```
namespace App\Models;
```

```
use App\Models\User;
```

```
use App\Models\Wallet;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use Config;
```

```
use Illuminate\Support\Facades\Log;
```

```
class WalletLogic extends Model
```

```
{
```

```
    /*
```

```
        WayForPay API
```

```
    */
```

```
    public static function AddCardOld($data, User $user)
```

```
//чистая
```

```
{
```

```
    $merchant = config('wayforpay.way_for_pay_merchant');
```

```
    $domain = config('wayforpay.way_for_pay_domain');
```

```
    $key = config('wayforpay.way_for_pay_key');
```

```
    $currency = 'UAH';
```

```
    $string =
```

```
$merchant.';'.$domain.';'.$data['order'].';'.$data['amount'].';'.$
currency;
```

```
    $signature = hash_hmac("md5",$string,$key);
```

```
    $fields = json_encode(["apiVersion" => 1,
```

```
    "transactionType" => "VERIFY",
```

```
    "merchantAccount" => $merchant,
```

```
    "merchantDomainName" => $domain,
```

```
    "merchantSignature" => $signature,
```

```
    "orderReference" => $data['order'],
```

```
    "amount" => $data['amount'],
```

```
    "currency" => $currency,
```

					КП.ІП-5121.045440.03.13	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "card" => $data['card'],
        "expMonth" => $data['expMonth'],
        "expYear" => $data['expYear'],
        "cardCvv" => $data['cvv'],
        "cardHolder" => $data['cardHolder']]);
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,
"https://api.wayforpay.com/api");
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json;charset=utf-8'));
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
    $resp = curl_exec($ch);
    curl_close($ch);
    $wallet = new Wallet();
    $wallet->user_id = $user->id;
    //remove this later
    if (isset($resp->recToken))
        $wallet->rec_token = $resp->recToken;
    else
        $wallet->rec_token = "Fish";
    $wallet->save();
    return json_encode($resp);
}

public static function AddCard($data, User $user) // через
вебвью
{
    $merchant = config('wayforpay.way_for_pay_merchant');
    $domain = config('wayforpay.way_for_pay_domain');
    $key = config('wayforpay.way_for_pay_key');
    $currency = 'UAH';

```

```

$string =
$merchant.';'.$domain.';'.$data['order'].';'.$data['amount'].';'.$
currency;

$signature = hash_hmac("md5",$string,$key);
$fields = json_encode([
    "apiVersion" => 1,
    "merchantAccount" => $merchant,
    "merchantDomainName" => $domain,
    "merchantSignature" => $signature,
    "orderReference" => $data['order'],
    "amount" => $data['amount'],
    "currency" => $currency]);
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,
"https://secure.wayforpay.com/verify");
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json;charset=utf-8'));
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
$resp = curl_exec($ch);
curl_close($ch);
$wallet = new Wallet();
$wallet->user_id = $user->id;
//remove this later
if (isset($resp->recToken))
    $wallet->rec_token = $resp->recToken;
else
    $wallet->rec_token = "Fish";
$wallet->save();
return $resp;
}

public static function getRecTokenFromWeb($data)

```

```

    }
    $userId = substr($data['orderReference'], 26);
    $wallet = Wallet::create([
        'user_id' => $userId,
        'rec_token' => $data['recToken'],
        'card_last_nums' => $data['cardPan'],
        'card_type' => $data['cardType']
    ]);
}
public static function Purchase()
{
    $merchant = config('wayforpay.way_for_pay_merchant');
    $domain = config('wayforpay.way_for_pay_domain');
    $key = config('wayforpay.way_for_pay_key');
    $currency = 'UAH';
    $data['productName'] = ["check"];
    $data['productPrice'] = ["1"];
    $data['productCount'] = ["1"];
    $data['orderDate'] = "1559313996";
    $data['order'] = "check";
    $data['amount'] = "1";
    $string =
$merchant.';'.$domain.';'.$data['order'].';'.$data['orderDate'].';
'.$data['amount'].';'.$currency.';'.$data['productName'][0].';'.$d
ata['productCount'][0].';'.$data['productPrice'][0];
    $signature = hash_hmac("md5",$string,$key);
    $fields = json_encode([
        "merchantAccount" => $merchant,
        "merchantDomainName" => $domain,
        "merchantSignature" => $signature,
        "orderReference" => $data['order'],
        "orderDate" => $data['orderDate'],
        "amount" => $data['amount'],
        "productName" => $data['productName'],

```

```

        "productPrice" => $data['productPrice'],
        "productCount" => $data['productCount'],
        "serviceUrl"                                     ==>
"https://consultant.net.ua/api/saveRecToken",
        "currency" => $currency]);
    $ch = curl_init();
    curl_setopt($ch,                                     CURLOPT_URL,
"https://secure.wayforpay.com/pay");
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json;charset=utf-8'));
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
    $resp = curl_exec($ch);
    curl_close($ch);
    Log::alert('Check Purchase' . $resp);
    return $resp;
}
public static function TransactFromCardToCard($data)
{
    $merchant = config('wayforpay.way_for_pay_merchant');
    $domain = config('wayforpay.way_for_pay_domain');
    $key = config('wayforpay.way_for_pay_key');
    $currency = 'UAH';
    //change this
    $recToken = "Fish";
    $string                                     =
$merchant.';'.$data['order'].';'.$data['amount'].';'.$currency.';'.
$.cardBeneficiary;
    $signature = hash_hmac("md5",$string,$key);

    $fields = json_encode(["apiVersion" => 1,
        "transactionType" => "P2P_TRANSFER",

```

```

"merchantAccount" => $merchant,
"merchantAuthType" => "SimpleSignature",
"merchantDomainName" => $domain,
"merchantSignature" => $signature,
"orderReference" => $data['order'],
"orderDate" => $data['orderDate'],
"amount" => $data['amount'],
"currency" => $currency,
"recToken" => $recToken,
"clientIpAddress" => $ipAddress,
"cardBeneficiary" => $cardBeneficiary,
]);
$ch = curl_init();
curl_setopt($ch,                                CURLOPT_URL,
"https://api.wayforpay.com/api");
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json;charset=utf-8'));
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
$res = curl_exec($ch);
curl_close($ch);
return json_encode($res);
}
public static function Charge($data)
{
    $merchant = config('wayforpay.way_for_pay_merchant');
    $domain = config('wayforpay.way_for_pay_domain');
    $key = config('wayforpay.way_for_pay_key');
    $currency = 'UAH';
    //change this
    $recToken = "Fish";

```

```

$string =
$merchant.';'.$domain.';'.$data['order'].';'.$data['orderDate'].';
'.$data['amount'].';'.$currency.';'.$data['productName'][0].';'.$d
ata['productCount'][0].';'.$data['productPrice'][0];
$signature = hash_hmac("md5",$string,$key);

$fields = json_encode(["apiVersion" => 1,
    "transactionType" => "CHARGE",
    "merchantAccount" => $merchant,
    "merchantTransactionType" => "SALE",
    "merchantTransactionSecureType" => "NON3DS",
    "merchantDomainName" => $domain,
    "merchantSignature" => $signature,
    "orderReference" => $data['order'],
    "orderDate" => $data['orderDate'],
    "amount" => $data['amount'],
    "currency" => $currency,
    "productName" => $data['productName'],
    "productPrice" => $data['productPrice'],
    "productCount" => $data['productCount'],
    "clientFirstName" => $data['clientFirstName'],
    "clientLastName" => $data['clientLastName'],
    "clientEmail" => $data['clientEmail'],
    "clientPhone" => $data['clientPhone'],
    "clientCountry" => $data['clientCountry'],
    "recToken" => $recToken,
]);
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,
"https://api.wayforpay.com/api");
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json;charset=utf-8'));
curl_setopt($ch, CURLOPT_POST, 1);

```

```

        curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
        $resp = curl_exec($ch);
        curl_close($ch);
        return json_encode($resp);
    }
    public static function TransactionFromAccountToCard($data)
    {
        $merchant = config('wayforpay.way_for_pay_merchant');
        $domain = config('wayforpay.way_for_pay_domain');
        $key = config('wayforpay.way_for_pay_key');
        $currency = 'UAH';
        //change this
        $recToken = "Fish";
        $rec2Token = "Fish";
        $string =
$merchant.';'.$data['order'].';'.$data['amount'].';'.$currency.';'.
.$data['rec2Token'];
        $signature = hash_hmac("md5",$string,$key);
        $fields = json_encode(["apiVersion" => 1,
            "transactionType" => "P2P_CREDIT",
            "merchantAccount" => $merchant,
            "merchantSignature" => $signature,
            "orderReference" => $data['order'],
            "amount" => $data['amount'],
            "currency" => $currency,
            "rec2Token" => $rec2Token,
        ]);
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL,
"https://api.wayforpay.com/api");
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json;charset=utf-8'));
    }

```



```

    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
    $resp = curl_exec($ch);
    curl_close($ch);

    return json_encode($resp);
}

public static function ShowTransactions($data)
{
    $merchant = config('wayforpay.way_for_pay_merchant');
    $domain = config('wayforpay.way_for_pay_domain');
    $key = config('wayforpay.way_for_pay_key');
    $string = $merchant.';'.$data['dateBegin'].';'.$data['dateEnd'];
    $signature = hash_hmac("md5",$string,$key);

    $fields = json_encode(["apiVersion" => 1,
    "transactionType" => "TRANSACTION_LIST",
    "merchantAccount" => $merchant,
    "merchantSignature" => $signature,
    "dateBegin" => $data['dateBegin'],
    "dateEnd" => $data['dateEnd']] );
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,
    "https://api.wayforpay.com/api");
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json;charset=utf-8'));
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
    $resp = curl_exec($ch);
    curl_close($ch);

```

```

        return json_encode($resp);
    }
    public static function ShowMainCardData($id)
    {
        $wallet = Wallet::select(
            'card_last_nums',
            'card_expired_month',
            'card_expired_year',
            'card_type'
        )
        ->where('user_id', $id)
        ->first();
        return $wallet;
    }
    /*
    Consultant Logic
    */

    /**
     * Add CArd Number for Lawyers
     * @param array $data [card number]
     * @param User $user
     */
    public static function AddCardNumber($data, User $user)
    {
        $wallet = Wallet::where('user_id', $user->id)->first();
        $wallet->card_number_trans = $data['card_number'];
        $wallet->save();
        return json_encode(["response" => 200]);
    }
    /**
     * Delete Card from user account
     * @param User $user
     */

```

```

public static function RemoveCardRecToken(User $user)
{
    $wallet = Wallet::where('user_id', $user->id)->first();
    $wallet->delete();
    return json_encode(["response" => 200]);
}
/**
 * Save main information about the card (last 4 digits,
month, year)
 * @param array $data [XXXX, MM, YYYY]
 * @param User $user
 */
public static function SaveMainCardData($data, User $user)
{
    $wallet = Wallet::where('user_id', $user->id)->first();
    $wallet->card_last_nums = $data['card_last_nums'];
    $wallet->card_expired_month = $data['card_expired_month'];
    $wallet->card_expired_year = $data['card_expired_year'];
    $wallet->card_type = $data['card_type'];
    $wallet->save();
    return json_encode(["response" => 200]);
}
}

```

Controllers

Api/AuthLoginController

<?php

namespace App\Http\Controllers\Api\Auth;

use App\Http\Controllers\Controller;

use App\Models\Client;

use App\Models\TURBOSMS;

use Illuminate\Foundation\Auth\AuthenticatesUsers;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Log;

use Illuminate\Support\Facades\Session;

```

use Illuminate\Http\Request;
use App\Models\User;
class LoginController extends Controller
{
    /**
     |-----
    -----
    | Login Controller
    |-----
    -----
    |
    | This controller handles authenticating users for the
application and
    | redirecting them to your home screen. The controller uses
a trait
    | to conveniently provide its functionality to your
applications.
    |
    */
    use AuthenticatesUsers;
    /**
     * Where to redirect users after login.
     *
     * @var string
     */
    protected $redirectToAfterLogout = '/';
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest', ['except' => 'logout']);
    }

```

```

    }
    public function login(Request $request)
    {
        $user = User::where('phone', $request->phone)->first();
        if (!$user)
        {
            return response()->json(['error' => 'Unauthorized'],
401);
        }
        $client = Client::where('user_id', $user->id)->first();
        $client->code_verification = rand(100000, 999999);
        if ($request->device_token != null) {
            $client->device_token = $request->device_token;
        }
        $client->save();
        //$turbosms = new TURBOSMS();
        //$result = $turbosms->SMSTurbo($request->phone, $client-
>code_verification);
        $data = [
            'status' => 0,
            'message' => 'verification required'
        ];
        return response()->json(["data" => $data]);
    }
    /**
     * Logout, Clear Session, and Return.
     *
     * @return void
     */
    public function logout()
    {
        $user = Auth::user();
        $user->activated = 0;
        $user->save();
    }

```

```

        Log::info('User Logged Out. ', [$user]);
        Auth::logout();
        Session::flush();
        return redirect(property_exists($this,
'redirectAfterLogout') ? $this->redirectAfterLogout : '/');
    }
}

```

Api/AuthRegisterController

```
<?php
```

```
namespace App\Http\Controllers\Api\Auth;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Models\Referral;
```

```
use App\Models\User;
```

```
use App\Traits\ActivationTrait;
```

```
use App\Traits\CaptchaTrait;
```

```
use App\Traits\CaptureIpTrait;
```

```
use Illuminate\Auth\Events\Registered;
```

```
use Illuminate\Foundation\Auth\RegistersUsers;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Hash;
```

```
use Illuminate\Support\Facades\Log;
```

```
use Illuminate\Support\Facades\Validator;
```

```
use jeremykenedy\LaravelRoles\Models\Role;
```

```
use App\Models\Client;
```

```
use App\Models\TURBOSMS;
```

```
class RegisterController extends Controller
```

```
{
```

```
    /*
```

```
    | -----
```

```
-----
```

```
    | Register Controller
```

```
    | -----
```

```
-----
```

```
    |
```

| This controller handles the registration of new users as well as their

| validation and creation. By default this controller uses a trait to

| provide this functionality without requiring any additional code.

```
|
*/
use ActivationTrait;
use CaptchaTrait;
use RegistersUsers;
/**
 * Where to redirect users after registration.
 *
 * @var string
 */
protected $redirectTo = '/activate';
/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guest', [
        'except' => 'logout',
    ]);
}
public function register(Request $request)
{
    $validator = Validator::make($request->all(), [
        'first_name' => 'required',
        'last_name' => 'required',
        'phone' => 'required|unique:users|max:12'
```

```

    });
    if ($validator->fails()) {
        return response()->json(['error'=>$validator->errors()],
401);
    }
    event(new Registered($user = $this->create($request-
>all())));
    $data = [
        'status' => 0,
        'message' => 'verification required'
    ];
    return response()->json(["data" => $data]);
}
/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 *
 * @return User
 */
protected function create(array $data)
{
    $ipAddress = new CaptureIpTrait();
    $role = Role::where('slug', '=', 'unverified')->first();
    $user = User::where('phone', $data['phone'])->first();
    if($user && $user->activated == 0)
    {
        $user->delete();
    }
    $user = User::create([
        'first_name'      => $data['first_name'],
        'last_name'       => $data['last_name'],
        'phone'           => $data['phone'],
        'password'        => Hash::make('password'),

```



```

        'token'                => str_random(64),
        'signup_ip_address' => $ipAddress->getClientIp(),
        'activated'          => !config('settings.activation')
    ]);
    $referralLinks = Referral::select('referral_url')->get();
    $hasRefferal = true;
    while ($hasRefferal) {
        $referral = str_random(12);
        $hasRefferal = false;
        foreach ($referralLinks as $rl) {
            if ($rl == $referral) {
                $hasRefferal = true;
            }
        }
    }
    $referralLink = Referral::create([
        'user_id'                => $user->id,
        'referral_url'           => $referral
    ]);
    $client = Client::create([
        'user_id'                => $user->id,
        'code_verification' => rand(100000, 999999)
    ]);
    if (array_key_exists('device_token', $data)) {
        $client->device_token = $data['device_token'];
        $client->save();
    }
    $user->attachRole($role);
    //$this->initiateEmailActivation($user);
    //$turbosms = new TURBOSMS();
    //$result = $turbosms->SMSTurbo($data['phone'], $client->code_verification);
    return $user;
}

```

```

public function verify(Request $request)
{
    $user = User::where('phone', $request->phone)->first();
    $client = Client::where('user_id', $user->id)->first();
    $code_verification = $client->code_verification;
    if ($code_verification == $request->code_verification)
    {
        $user->activated = 1;
        $user->save();
        $credentials = [
            "phone" => $request->phone,
            "password" => 'password'
        ];
        if (!$token = auth('api')->attempt($credentials)) {
            return response()->json(['error' => 'Unauthorized'],
401);
        }
        return User::respondWithToken($token);
    } else {
        $language = User::select('language')->where('phone',
$request->phone)->first()->language;
        switch ($language) {
            case 'ru':
                $message = 'Ошибка. Пожалуйста попробуйте еще раз';
                break;
            case 'uk':
                $message = 'Помилка. Будь ласка спробуйте ще раз';
                break;
            case 'en':
                $message = 'Error. Please try again';
                break;
        }
        return response()->json([
            'error' => 'Unauthorized',

```

```

        'message' => $message], 401);
    }
}
public function resendCode(Request $request)
{
    $user = User::where('phone', $request->phone)->get();
    if (!$user)
    {
        return response()->json(['error' => 'No user. Please
register again'], 401);
    }
    $client = Client::where('user_id', $user->id)->first();
    $client->code_verification = rand(100000, 999999);
    $client->save();
    $turbosms = new TURBOSMS();
    $result = $turbosms->SMSTurbo($request->phone, $client-
>code_verification);
    return response()->json(['data' => $user], 200);
}
public function SMSTurbo()
{
    $client = new TURBOSMS();
    $result = $client->SMSTurbo('380668348444', '456456');
    return response()->json([$result]);
}
}

```

Api/Auth/ResetPhoneController

```

<?php
namespace App\Http\Controllers\Api\Auth;
use App\Models\User;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

```

```

use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Facades>Password;
use App\Mail\ResetPhone;
class ResetPhoneController extends Controller
{
    public function reset(Request $request)
    {
        $credentials = $request->only(['email']);
        $user = User::where('email', $request->email)->first();
        if (!$user) {
            return response()->json([
                'error' => 'no such user with this email',
                'status' => 0
            ], 401);
        }
        $response = $this->broker()->sendResetLink(
            $request->only('email')
        );
        $data = [
            'first_name' => $user->first_name,
            'middle_name' => $user->middle_name,
            'email'      => $user->email,
            'token'      => $user->token
        ];
        Mail::to($user->email)->send(new ResetPhone($data));
        return $this->sendResetLinkResponse($request, $response);
    }
}
/**
 * Get the response for a successful password reset link.
 *
 * @param  \Illuminate\Http\Request  $request
 * @param  string  $response

```

```

*                                                                 @return
\Illuminate\Http\RedirectResponse|\Illuminate\Http\JsonResponse
*/
protected function sendResetLinkResponse(Request $request,
$response)
{
    return response()->json(['status', trans($response)]);
}
/**
 * Get the response for a failed password reset link.
 *
 * @param \Illuminate\Http\Request $request
 * @param string $response
 *
 *                                                                 @return
\Illuminate\Http\RedirectResponse|\Illuminate\Http\JsonResponse
*/
protected function sendResetLinkFailedResponse(Request
$request, $response)
{
    return response()->json(['email' => trans($response)]);
}
/**
 * Get the broker to be used during password reset.
 *
 * @return \Illuminate\Contracts\Auth\PasswordBroker
 */
public function broker()
{
    return Password::broker();
}
public function showFormRestorePhone($token){
    $data = [
        'token' => $token,
    ];

```

```

        return view('auth.phones.reset')->with($data);
    }
    public function restorePhone(Request $request){
        $user = User::where('token', $request->token)->first();
        if ($user->email === $request->email &&
Hash::check($request->password, $user->password) && $user->phone
=== $request->old_phone_number) {
            $user->phone = $request->new_phone_number;
            $user->save();
        } else {
            return response()->json(['WRONG CREDENTIALS']);
        }
        return view('welcome');
    }
}

```

Api/AnswerController

<?php

```

namespace App\Http\Controllers\Api\Auth;
use App\Models\User;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Facades>Password;
use App\Mail\ResetPhone;
class ResetPhoneController extends Controller
{
    public function reset(Request $request)
    {
        $credentials = $request->only(['email']);
        $user = User::where('email', $request->email)->first();
        if (!$user) {

```

```

        return response()->json([
            'error' => 'no such user with this email',
            'status' => 0
        ], 401);
    }
    $response = $this->broker()->sendResetLink(
        $request->only('email')
    );
    $data = [
        'first_name' => $user->first_name,
        'middle_name' => $user->middle_name,
        'email'      => $user->email,
        'token'      => $user->token
    ];
    Mail::to($user->email)->send(new ResetPhone($data));
    return $this->sendResetLinkResponse($request, $response);
}

/**
 * Get the response for a successful password reset link.
 *
 * @param \Illuminate\Http\Request $request
 * @param string $response
 *
 * @return \Illuminate\Http\RedirectResponse|\Illuminate\Http\JsonResponse
 */
protected function sendResetLinkResponse(Request $request,
$response)
{
    return response()->json(['status', trans($response)]);
}

/**
 * Get the response for a failed password reset link.
 *
 * @param \Illuminate\Http\Request $request

```

```

    * @param string $response
    *
    * @return
    \Illuminate\Http\RedirectResponse|\Illuminate\Http\JsonResponse
    */
    protected function sendResetLinkFailedResponse(Request
$request, $response)
    {
        return response()->json(['email' => trans($response)]);
    }
    /**
    * Get the broker to be used during password reset.
    *
    * @return \Illuminate\Contracts\Auth\PasswordBroker
    */
    public function broker()
    {
        return Password::broker();
    }
    public function showFormRestorePhone($token){
        $data = [
            'token' => $token,
        ];
        return view('auth.phones.reset')->with($data);
    }
    public function restorePhone(Request $request){
        $user = User::where('token', $request->token)->first();
        if ($user->email === $request->email &&
Hash::check($request->password, $user->password) && $user->phone
=== $request->old_phone_number) {
            $user->phone = $request->new_phone_number;
            $user->save();
        } else {
            return response()->json(['WRONG CREDENTIALS']);
        }
    }

```



```

        return view('welcome');
    }
}

```

Api/ProfilesController

```

<?php
namespace App\Http\Controllers\Api;
use App\Models\Advocate;
use App\Models\Answer;
use App\Models\Booker;
use App\Models\BookerCompany;
use App\Models\Client;
use App\Models\LawCompany;
use App\Models\Lawer;
use App\Models\Profile;
use App\Models\Referral;
use App\Models\Theme;
use App\Models\Topic;
use App\Models\User;
use App\Traits\CaptureIpTrait;
use Validator;
use Illuminate\Database\Eloquent\ModelNotFoundException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Input;
use Illuminate\Support\Facades\Log;
use jeremykenedy\Uuid\Uuid;
class ProfilesController extends Controller
{
    /**
     * Create a new controller instance.

```

```

*
* @return void
*/
/*public function __construct()
{
    $this->middleware('auth');
}*/
/**
* Get a validator for an incoming registration request.
*
* @param array $data
*
* @return \Illuminate\Contracts\Validation\Validator
*/
public function profile_validator(array $data)
{
    return Validator::make($data, [
        'theme_id'          => '',
        'location'          => '',
        'bio'                => 'max:500',
        'twitter_username' => 'max:50',
        'github_username'  => 'max:50',
        'avatar'            => '',
        'avatar_status'     => '',
    ]);
}
public function user_validator(array $data)
{
    return Validator::make($data, [
        'first_name'        => 'required',
        'last_name'         => 'required',
        'city'              => '',
        'email'              => 'email|unique:users',
        'password'          => 'max:50',
    ]);
}

```

```

        'password_confirmation' => 'same:password'
    });
}
/**
 * Fetch user
 * (You can extract this to repository method).
 *
 * @param $username
 *
 * @return mixed
 */
public function getUserByUsername($username)
{
    return User::with('profile')->whereName($username)-
>firstOrFail();
}
public function getUserById($id)
{
    return User::with('profile')->where('id', $id)-
>firstOrFail();
}
/**
 * Show my profile.
 *
 * @param string $username
 *
 * @return Response
 */
public function showProfile()
{
    try {
        $user = Auth::user();
    } catch (ModelNotFoundException $exception) {

```

```

        return response()->json(['error' => 'Unauthorized'],
401);
    }
    $clientInfo = Client::where('user_id' , $user->id)-
>first();
    $user->freeQuestions = $clientInfo->free_questions;
    return response()->json(["data" => [$user]]);
}
public function showReferralCode()
{
    $user = Auth::user();
    $referralLink = Referral::where('user_id' , $user->id)-
>first();
    return response()->json(["data" => [$referralLink]]);
}
/**
 * Set Notifications for user.
 *
 * @throws Laracasts\Validation\FormValidationException
 *
 * @return mixed
 */
public function setNotifications(Request $request)
{
    try {
        $user = Auth::user();
    } catch (ModelNotFoundException $exception) {
        abort(404);
    }
    $user->notification_client = $request->notificationClient
? true :false;
    $user->notification_system = $request->notificationSystem
? true :false;
    $ipAddress = new CaptureIpTrait();

```

```

        $user->updated_ip_address = $ipAddress->getClientIp();
        $user->save();
        return response()->json(["data" => ["success"]]);
    }
    /**
     * Update a user's profile.
     *
     * @param $username
     *
     * @throws Laracasts\Validation\FormValidationException
     *
     * @return mixed
     */
    public function setProfile(Request $request)
    {
        $user = Auth::user();
        $ipAddress = new CaptureIpTrait();
        $user_validator = $this->user_validator($request->all());
        if ($user_validator->fails()) {
            return response()->json(['error' => 'Validation fails'],
403);
        }
        $user->first_name = $request->first_name;
        $user->last_name = $request->last_name;
        $user->middle_name = $request->middle_name;
        $user->email = $request->email;
        $user->city = $request->city;
        if ($request->password) {
            $user->password = Hash::make($request->password);
        }
        $user->updated_ip_address = $ipAddress->getClientIp();
        $user->save();
        $clientInfo = Client::where('user_id' , $user->id)-
>first();
    
```

```

        $user->freeQuestions = $clientInfo->free_questions;
        return response()->json(["data" => [$user]]);
    }
    public function uploadImage(Request $request)
    {
        $user = Auth::user();
        $ipAddress = new CaptureIpTrait();
        if ($request->hasFile('avatar')) {
            if ($request->file('avatar')->isValid()) {
                $extension          =          $request->file('avatar')->getClientOriginalExtension();
                $filename = 'avatar.'.$extension ;
                $save_path  =  public_path().'/assets/users/id/'.$user->id.'/uploads/avatar/';
                $path        =          '/assets/users/id/'.$user->id.'/uploads/avatar/'.$filename;
                $request->file('avatar')->move($save_path, $filename);
                $user->avatar = $path;
            }
        }
        $user->updated_ip_address = $ipAddress->getClientIp();
        $user->save();
        return response()->json(["data" => [$user]]);
    }
    public function showConsultantProfile(Request $request) {
        $user = Auth::user();
        $consultant = User::select(
            'id',
            'first_name',
            'last_name',
            'middle_name',
            'type',
            'email',
            'avatar',

```

```

        'city',
        'phone',
        'type',
        'activated'
    )
    ->where('id', $request->user_id)
    ->first();
    if (!$consultant->activated) {
        return response()->json([ "data" => [
            'activated' => false
        ]]);
    }
    switch ($consultant->type) {
        case 'advocate' :
            $advocate = Advocate::where('user_id', $consultant->id)->first();
            $consultant->calling_time = $advocate->calling_time;
            $consultant->working_time = $advocate->working_time;
            Log::alert("and was here");
            break;
        case 'jurist' :
            $lawer = Lower::where('user_id', $consultant->id)->first();
            $consultant->calling_time = $lawer->calling_time;
            $consultant->working_time = $lawer->working_time;
            break;
        case 'booker' :
            $booker = Booker::where('user_id', $consultant->id)->first();
            $consultant->calling_time = $booker->calling_time;
            $consultant->working_time = $booker->working_time;
            break;
        case 'bookercompany' :

```

```

        $bookercompany = BookerCompany::where('user_id',
$consultant->id)->first();
        $consultant->company_name = $bookercompany-
>company_name;
        $consultant->occupation = $bookercompany->occupation;
        $consultant->calling_time = $bookercompany-
>calling_time;
        $consultant->working_time = $bookercompany-
>working_time;
        break;
    case 'lawcompany' :
        $lawcompany = LawCompany::where('user_id', $consultant-
>id)->first();
        $consultant->company_name = $lawcompany->company_name;
        $consultant->occupation = $lawcompany->occupation;
        $consultant->calling_time = $lawcompany->calling_time;
        $consultant->working_time = $lawcompany->working_time;
        break;
    }
    $consultant->answers = count(Answer::where('user_id',
$consultant->id)->get());
    $consultant->bestAnswers = count(Answer::where('user_id',
$consultant->id)->where('best_answer', true)->get());
    $consultant->topic = Topic::select('name')->where('type',
$consultant->type)->first()->name;
    return response()->json([ "data" => [$consultant]]);
}
public function changeLanguage(Request $request)
{
    try {
        $user = Auth::user();
    } catch (ModelNotFoundException $exception) {
        abort(404);
    }
}

```



```

        $user->language = $request->language;
        $ipAddress = new CaptureIpTrait();
        $user->updated_ip_address = $ipAddress->getClientIp();
        $user->save();
        return response()->json(["data" => ["success"]]);
    }
    public function setDeviceToken(Request $request)
    {
        try {
            $user = Auth::user();
        } catch (ModelNotFoundException $exception) {
            abort(404);
        }
        $client = Client::where('user_id', $user->id)->first();
        $client->device_token = $request->device_token;
        $client->save();
        return response()->json(["data" => ["success"]]);
    }
    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int                        $id
     *
     * @return \Illuminate\Http\Response
     */
    public function deleteUserAccount(Request $request, $id)
    {
        $currentUser = \Auth::user();
        $user = User::findOrFail($id);
        $ipAddress = new CaptureIpTrait();
        $validator = Validator::make($request->all(),
            [
                'checkConfirmDelete' => 'required',

```

```

],
[
    'checkConfirmDelete.required' =>
trans('profile.confirmDeleteRequired'),
]
);
if ($user->id != $currentUser->id) {
    return redirect('profile/'.$user->name.'/edit')-
>with('error', trans('profile.errorDeleteNotYour'));
}
if ($validator->fails()) {
    return back()->withErrors($validator)->withInput();
}
// Create and encrypt user account restore token
$sepKey = $this->getSeperationKey();
$userIdKey = $this->getIdMultiKey();
$restoreKey = config('settings.restoreKey');
$encrypter = config('settings.restoreUserEncType');
$level1 = $user->id * $userIdKey;
$level2 = urlencode(Uuid::generate(4).$sepKey.$level1);
$level3 = base64_encode($level2);
$level4 = openssl_encrypt($level3, $encrypter,
$restoreKey);
$level5 = base64_encode($level4);
// Save Restore Token and Ip Address
$user->token = $level5;
$user->deleted_ip_address = $ipAddress->getClientIp();
$user->save();
// Send Goodbye email notification
$this->sendGoodbyeEmail($user, $user->token);
// Soft Delete User
$user->delete();
// Clear out the session
$request->session()->flush();

```

```

        $request->session()->regenerate();
        return redirect('/login/')->with('success',
trans('profile.successUserAccountDeleted'));
    }
    public function changeOnModerator(Request $request)
    {
        try {
            $user = Auth::user();
        } catch (ModelNotFoundException $exception) {
            abort(404);
        }
    }
}

```

Api/QuestionController

```

<?php
namespace App\Http\Controllers\Api;
use App\Events\QuestionAppeared;
use App\Http\Resources\TopicResource;
use App\Models\Answer;
use App\Models\Topic;
use App\Models\User;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Resources\QuestionResource;
use App\Models\Question;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Validator;
class QuestionController extends Controller
{
    public function showTopics($lang)
    {
        $topics = Topic::all();
    }
}

```

```

switch ($lang)
{
    case 'en':
        $topics[0]->name = 'Advocate';
        $topics[1]->name = 'Jurist';
        $topics[2]->name = 'Accountant';
        $topics[3]->name = 'Accounting company';
        $topics[4]->name = 'Law company';
        break;
    case 'uk':
        $topics[3]->name = 'Бухгалтерська компанія';
        $topics[4]->name = 'Юридична компанія';
    }
    return TopicResource::collection($topics);
}
/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    $user = Auth::user();
    $questions = Question::with('topic')->with('answer')->with(array('answer.user'=>function($query){
        $query->select('id', 'first_name', 'last_name', 'avatar');
    }))->where('user_id', $user->id)->get();
    return QuestionResource::collection($questions);
}
/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request

```

```

* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    $question = Question::where('user_id', $request->user()->id)->where('closed', false)->first();
    $language = User::select('language')->where('id', $request->user()->id)->first()->language;
    if ($question)
    {
        switch ($language) {
            case 'ru':
                $message = 'Вы не можете задать новый вопрос, так как у вас уже есть один активный вопрос';
                break;
            case 'uk':
                $message = 'Ви не можете задати нове запитання, так як у вас ще є одне активне запитання';
                break;
            case 'en':
                $message = 'You cant create new question, because you have not closed questions';
                break;
        }
        return response()->json([
            "data" =>
                ['status' => 0,
                 'message' => $message
                ]
        ]);
    }
    $validator = Validator::make($request->all(), [
        'title' => 'required|max:50|min:3',
        'question' => 'required|max:250|min:10',
    ]);
}

```

```

        if ($validator->fails()) {
            switch ($language) {
                case 'ru':
                    $message = 'Слишком короткий итл слишком длинный
текст вопроса';
                    break;
                case 'uk':
                    $message = 'Помилка, дуже короткий або дуже довгий
заголовок';
                    break;
                case 'en':
                    $message = 'Error. The title is either too short or
too long';
                    break;
            }
            return response()->json([
                'error'=>$validator->errors(),
                'message' => $message
            ], 401);
        }
        $question = Question::create([
            'user_id' => $request->user()->id,
            'topic_id' => $request->topic_id,
            'title'    => $request->title,
            'question' => $request->question,
        ]);
        $question->first_name = $question->user->first_name;
        $question->last_name = $question->user->last_name;
        $question->city = $question->user->city;
        event(new QuestionAppeared($question));
        return new QuestionResource($question);
    }

    /**
     * Display the specified resource.

```

```

*
* @param Question $question
* @return \Illuminate\Http\Response
*/
public function show($id)
{
    $user = Auth::user();
    $question = Question::where('id', $id)->with('topic')-
>with('answer')->first();
    foreach ($question->answer as $answer) {
        $user = User::where('id', $answer->user_id)-
>with('profile')->first();
        $answers = Answer::where('user_id', $user->id)->get();
        $best_answers = Answer::where('user_id', $user->id)-
>where('best_answer', 1)->get();
        $answer->user = [
            "id" => $user->id,
            "first_name" => $user->first_name,
            "last_name" => $user->last_name,
            "middle_name" => $user->middle_name,
            "city" => $user->city,
            "avatar" => $user->avatar,
            "answers" => count($answers),
            "best_answers" => count($best_answers)
        ];
    }
    $question->first_name = $user->first_name;
    $question->last_name = $user->last_name;
    $question->city = $user->city;
    return response()->json(["data" => [$question]]);
}
/**
* Update the specified resource in storage.
*

```

```

* @param \Illuminate\Http\Request $request
* @param int $id
* @return \Illuminate\Http\Response
*/
public function update(Request $request, Question $question)
{
    // check if currently authenticated user is the owner of
the question
    if ($request->user()->id !== $question->user_id) {
        return response()->json(['error' => 'You can only edit
your own questions.'], 403);
    }
    $question->update($request->only(['question',
'topic_id']));
    return new QuestionResource($question);
}
/**
* Remove the specified resource from storage.
*
* @param int $id
* @return \Illuminate\Http\Response
*/
public function destroy(Question $question)
{
    $question->delete();
    return response()->json(null, 204);
}
}

```

Api/WalletController

<?php

namespace App\Http\Controllers\Api;

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-5121.045440.03.13 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 42 |


```

use App\Models\Wallet;
use DateTime;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Models\WalletLogic;
use Auth;
use Config;
use Illuminate\Support\Facades\Log;
class WalletController extends Controller
{
    public function addCard(Request $request)
    {
        $user = auth('api')->user();
        $freezeAmount = 1;
        $data = [
            "card"          => $request->card,
            "expMonth"      => $request->expMonth,
            "expYear"       => $request->expYear,
            "cardHolder"    => $request->cardHolder,
            "cvv"           => $request->cvv,
            "order"         => "Fish".str_random(8),
            "amount"        => $freezeAmount,
        ];
        $responseAddCard = WalletLogic::AddCard($data, $user);
        $data = [
            "card_last_nums"    => substr($request->card, -4),
            "card_expired_month" => $request->expMonth,
            "card_expired_year" => $request->expYear,
            "card_type"         => $request->card_type
        ];
        $responseAddMainCardInfo =
        WalletLogic::SaveMainCardData($data, $user);
        return response()->json([
            'responseAddCard'      => $responseAddCard,

```

```

        'responseAddMainCardInfo' => $responseAddMainCardInfo,
    ]);
}
public function getRecTokenUrl(Request $request)
{
    $data = [
        'orderReference' => $request->orderReference,
        'phone' => $request->phone,
        'createdDate' => $request->createdDate,
        'cardPan' => $request->cardPan,
        'cardType' => $request->cardType,
        'recToken' => $request->recToken,
        'transactionStatus' => $request->transactionStatus,
        'reasonCode' => $request->reasonCode
    ];
    if ($request->reasonCode == '1100')
    {
        $wallet = WalletLogic::getRecTokenFromWeb($data);
    }
}
public function purchase(Request $request)
{
    $responseAddMainCardInfo = WalletLogic::Purchase();
    return response()->json([$responseAddMainCardInfo]);
}
public function removeCard(Request $request)
{
    $user = auth('api')->user();
    $responseRemoveCard =
    WalletLogic::RemoveCardRecToken($user);
    return response()->json(['data' => $responseRemoveCard]);
}
public function showCardCred()
{

```

```

$user = auth('api')->user();
$wallet = WalletLogic::ShowMainCardData($user->id);
$merchant = config('wayforpay.way_for_pay_merchant');
$domain = config('wayforpay.way_for_pay_domain');
$key = config('wayforpay.way_for_pay_key');
$data['merchantAccount'] = $merchant;
$data['merchantDomainName'] = $domain;
$data['serviceUrl'] =
'https://consultant.net.ua/api/saveRecToken';
$data['currency'] = 'UAH';
$data['productName'] = ["cardVerification"];
$data['productPrice'] = ["1"];
$data['productCount'] = ["1"];
$date = new DateTime();
$data['orderDate'] = $date->getTimestamp();
$data['orderReference'] = "cardVerification" .
$data['orderDate'] . $user->id;
$data['amount'] = "1";
$string =
$merchant.';'.$domain.';'.$data['orderReference'].';'.$data['order
Date'].';'.$data['amount'].';'.$data['currency'].';'.$data['produc
tName'][0].';'.$data['productCount'][0].';'.$data['productPrice'][
0];

$signature = hash_hmac("md5",$string,$key);
$showData = [
    'user' => $user,
    'wallet' => $wallet,
    "data" => $data,
    'signature' => $signature
];
return response()->json(['data' => $showData]);
}

public function showCardForm()
{

```

```

        $user = auth('api')->user();
        $wallet = WalletLogic::ShowMainCardData($user->id);
        $data['merchant'] =
config('wayforpay.way_for_pay_merchant');
        $data['domain'] = config('wayforpay.way_for_pay_domain');
        $key = config('wayforpay.way_for_pay_key');
        $data['currency'] = 'UAH';
        $data['productName'] = ["cardVerification"];
        $data['serviceUrl'] =
'https://consultant.net.ua/api/saveRecToken';
        $data['productPrice'] = ["1"];
        $data['productCount'] = ["1"];
        $date = new DateTime();
        $data['orderDate'] = $date->getTimestamp();
        $data['order'] = "cardVerification" . $data['orderDate'] .
$user->id;
        $data['amount'] = "1";
        $string =
        $data['merchant'].';'.$data['domain'].';'.$data['order'].';'.$data
['orderDate'].';'.$data['amount'].';'.$data['currency'].';'.$data[
'productName'][0].';'.$data['productCount'][0].';'.$data['productP
rice'][0];
        $signature = hash_hmac("md5",$string,$key);
        $showData = [
            'user' => $user,
            'wallet' => $wallet,
            "data" => $data,
            'signature' => $signature
        ];
        return view('pages.account.card-mobile')->with($showData);
    }
}

```

КПІ.ІП-5121.045440.03.13

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-5121.045440.03.13 | Арк. |
| | | | | | | 47 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

ОНЛАЙН – ПЛАТФОРМА ДЛЯ НАДАННЯ ЮРИДИЧНИХ ПОСЛУГ

Програма та методика тестування

КПІ.ІІ-5121.045440.04.51

“ПОГОДЖЕНО”

Керівник проекту:

_____ І. П. Муха

Нормоконтроль:

_____ К.І.Ліщук

Виконавець:

_____ О.В. Субцельний

Київ – 2019 року

ЗМІСТ

| | |
|---|----------|
| 1 ОБ’ЄКТ ВИПРОБУВАНЬ..... | 3 |
| 2 МЕТА ТЕСТУВАННЯ | 3 |
| 3 МЕТОДИ ТЕСТУВАННЯ..... | 3 |
| 4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ | 4 |

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-5121.045440.04.51 | Арк. |
| | | | | | | 2 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Онлайн-платформа для надання юридичних послуг, яка являє собою Веб-сервіс, REST API для мобільних платформ, мобільний додаток на iOS та Android, створена на платформі Laravel та Node.js.

2 МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- функціональна працездатність елементів сторінок веб-сервісу;
- функціональна працездатність елементів сторінок мобільного додатку;
- успішна інтеграція платформи з сторонніми сервісами;
- забезпечення належного рівня безпеки даних;
- зручність роботи з веб-сервісом;
- зручність роботи з мобільним додатком;
- відповідність дизайну вимогам технічного завдання.

3 МЕТОДИ ТЕСТУВАННЯ

Тестування виконується методом Gray Box Testing. Перевіряється як код, так і безпосередньо програмний продукт на відповідність функціональним вимогам. Тестування відбувається на рівні «системного тестування».

Димне тестування буде запускатися кожного разу перед деплоєм нової версії на Debug конфігурації, та після деплоя на Release. У випадку Release, усі тести будуть проводитись в окремій базі даних.

Використовуються наступні методи:

- тестування конвертації та обміну даними;
- тестування зручності інтерфейсу;
- тестування інтерфейсу користувача;

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.04.51 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

- тестування функціоналу сайту та додатку (системне тестування);
- тестування навантаження;
- тестування безпеки;
- тестування стабільності.

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Обладнання:

- частина ПЗ, що розташована на сервері має бути протестована на сервері. Вимоги до сервера вказані у Технічному завданні;
- частина ПЗ, що розташована на клієнтській машині має бути протестована на клієнтській машині. Вимоги до клієнтської машини вказані у Технічному завданні.

Працездатність веб-ресурсу перевіряється шляхом:

- динамічного ручного тестування — введенням граничних та недопустимих значень в поля, які можна редагувати;
- динамічного ручного тестування на відповідність функціональним вимогам;
- статичного тестування коду;
- тестування при максимальному навантаженні;
- тестування стабільності роботи при різних умовах;
- тестування зручності використання;
- тестування інтерфейсу.

Працездатність мобільного додатку перевіряється шляхом:

- динамічного ручного тестування — введенням граничних та недопустимих значень в поля, які можна редагувати;
- динамічного ручного тестування на відповідність функціональним вимогам;
- тестування стабільності роботи при різних умовах;
- тестування зручності використання;

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.04.51 | Арк. |
| | | | | | | 4 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- тестування інтерфейсу.

Працездатність інтеграції сторонніх сервісів перевіряється шляхом:

- динамічного ручного тестування на відповідність функціональним вимогам;
- тестування стабільності роботи при різних умовах.

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

ОНЛАЙН – ПЛАТФОРМА ДЛЯ НАДАННЯ ЮРИДИЧНИХ ПОСЛУГ

Керівництво користувача

КПІ.ІП-5121.045440.05.34

“ПОГОДЖЕНО”

Керівник проекту:

_____ І.П. Муха

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ О.В. Субцельний

Київ – 2019 року

| | | | | | | |
|------|------|----------|--------|------|-------------------------------------|------|
| | | | | | <div>КПІ.ІП-5121.045440.03.13</div> | Арк. |
| | | | | | | 87 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ЗМІСТ

| | |
|--|----|
| Перелік умовних позначень та скорочень..... | 3 |
| Вступ..... | 4 |
| 1 ЧАСТИНА А Інструкція Клієнта Мобільний додаток iOS & Android.. | 5 |
| 1.1 Вимоги до апаратно – програмного забезпечення..... | 5 |
| 1.2 Початок роботи | 5 |
| 1.2.1 Вхід до Системи..... | 5 |
| 1.2.2 Перегляд та редагування профіля..... | 9 |
| 1.2.3 Налаштування..... | 10 |
| 1.2.4 Програма лояльності..... | 12 |
| 1.2.5 Питання..... | 13 |
| 2 ЧАСТИНА Б Інструкція Консультанта Веб-сервіс..... | 16 |
| 2.1 Вимоги до апаратно – програмного забезпечення..... | 16 |
| 2.2 Початок роботи..... | 16 |
| 2.2.1 Вхід до системи..... | 16 |
| 2.2.2 Профіль..... | 20 |
| 2.2.3 Налаштування..... | 21 |
| 2.2.4 Питання - відповіді..... | 23 |
| 3 ЧАСТИНА В Інструкція Модератора Веб-сервіс..... | 26 |
| 3.1 Вимоги до апаратно – програмного забезпечення..... | 26 |
| 3.2 Початок роботи..... | 26 |
| 3.2.1 Вхід до системи..... | 26 |
| 3.2.2 Керування користувачами..... | 27 |
| 4 ЧАСТИНА Г Інструкція Адміністратора Веб-сервіс..... | 30 |
| 4.1 Вимоги до апаратно – програмного забезпечення..... | 30 |
| 4.2 Початок роботи..... | 30 |
| 4.2.1 Вхід до системи..... | 30 |
| 4.2.2 Керування користувачами..... | 30 |

Перелік умовних позначень та скорочень

В Інструкції не використані скорочення та умовні позначення.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 3 |

ВСТУП

Онлайн-платформа для надання юридичних послуг "Консультант" надає можливість консультації з юристом, адвокатом, бухгалтером в онлайн-режимі. Основна функція і завдання онлайн-платформи - зв'язати клієнта і юриста / адвоката / бухгалтера для надання допомоги клієнту при виникненні будь-яких питань, пов'язаних з юриспруденцією.

Клієнт може задати будь-яке питання і отримати на нього відповідь від кількох виконавців, зареєстрованих в базі. Допускається необмежена кількість відповідей на питання клієнта, тому клієнт зобов'язаний вибрати одну з них, внаслідок чого у нього спишеться певна сума з банківської картки, яку він прив'язує при реєстрації на сервісі. Якщо клієнт не вибере відповідь, він не може ставити наступне питання.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 4 |

1 ЧАСТИНА А

Інструкція Клієнта

Мобільний додаток iOS & Android

1.1 Вимоги до апаратно – програмного забезпечення

Вимоги до апаратно-програмного забезпечення:

а) допустимі наступні версії ОС:

- iOS 12,11, 10, 9;
- Android 4.4 KitKat до Android 9.0 Pie.

б) додаток адаптований під наступні моделі телефонів:

- iPhone XS Max, XS, XR, X, 8, 8+, 7, 7+, 6, 6+, 6S, 6S+, 5C, 5S, SE, 5;
- також останні телефони Android;

в) додаток не адаптовано для роботи на iPad.

1.2 Початок роботи з додатком

Для роботи з сервісом необхідно вміти працювати з мобільними додатками.

1.2.1 Вхід до Системи

Для входу до сервісу “Консультант”, необхідно необхідновідкрити мобільний додаток. На екрані телефону з'явиться вікно з описом додатку та кнопкою “Далі” та “Пропустити”. Якщо користувач має потребу прочитати короткий опис програми, він має можливість натиснути кнопку “Далі” та пролистати слайди з описом програми. Якщо користувач не бажає читати опис, він має можливість пропустити, натиснувши кнопку “Пропустити”. Приклади наведені на скріншотах нижче.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 5 |

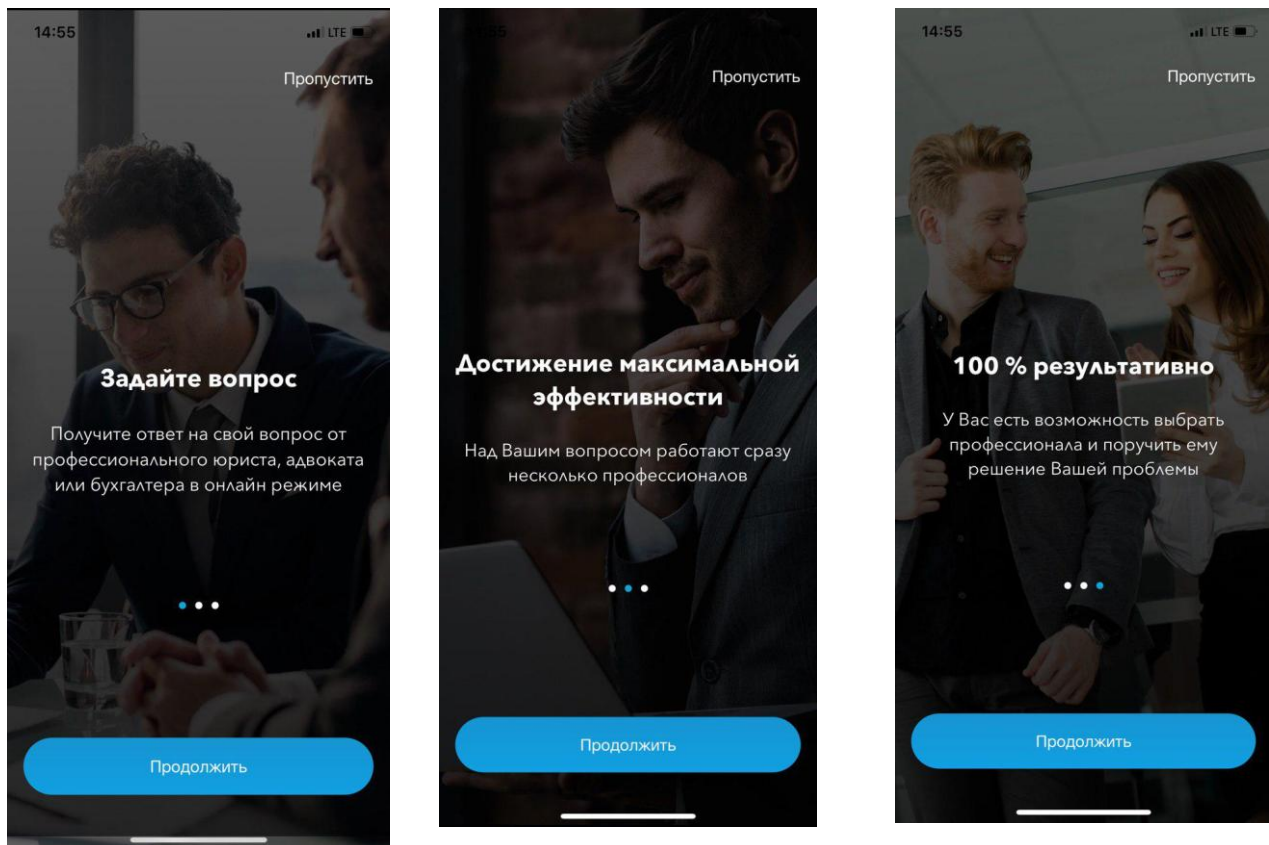


Рисунок 1.1 - Скріншоти вхідних екранів

Після слайдів продемонстрованих вище користувач потрапляє у головне вікно входу. Приклад головного вікна входу продемонстровано на малюнку.



Рисунок 1.2 - Скріншот головного екрану

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 6 |

Користувач має можливість зареєструватись як новий клієнт або авторизуватись у системі.

1.2.1.1 Реєстрація

Для реєстрації на платформі потрібно:

- а) натиснути кнопку “Зареєструватись”, що розташована під кнопкою “Увійти”;
- б) ввести свої персональні дані у формі, що з’явиться після натиснення кнопки “Зареєструватись”, а саме, ім’я, прізвище та номер телефона;
- в) натиснути кнопку “Зареєструватись”;
- г) отримати смс-код підтвердження, що прийде на номер телефона, вказаний у формі;
- д) ввести отриманий код-пароль у форму, що з’явиться після натиснення кнопки “Зареєструватись”;
- е) натиснути кнопку “Далі”.

Скріншоти описаних дій наведені нижче.

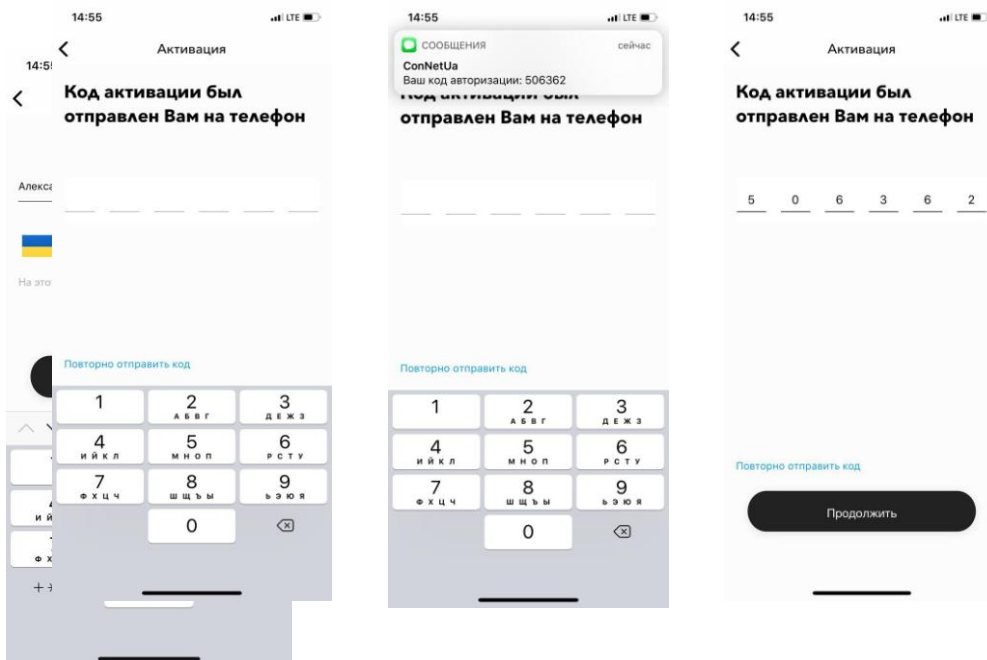


Рисунок 1.3 - Скріншоти екранів реєстрації

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 7 |

1.2.1.2 Авторизація

Для авторизації на платформі потрібно:

- а) натиснути кнопку “Увійти”;
 - б) ввести номер телефону в форму;
 - в) натиснути “Далі”;
 - г) отримати смс-повідомлення з кодом підтвердження;
 - д) ввести код в поле для вводу форми, що з'явиться після натиснення кнопки “Далі”;
 - е) натиснути кнопку “Далі”.
- Скріншоти описаний дій наведені нижче.

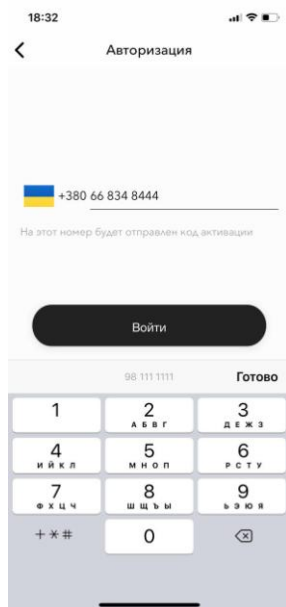


Рисунок 1.4 - Скріншоти екранів авторизації

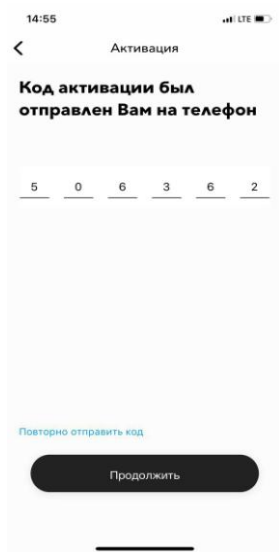


Рисунок 1.5 - Скріншоти екранів авторизації

1.2.2 Перегляд та редагування профіля

1.2.2.1 Переглянути профіль

Щоб переглянути профіль потрібно:

- а) бути авторизованим у мобільному додатку;
- б) перейти на вкладку профіля (значок чоловічка зправа знизу мобільного додатку).

Скріншоти наведені нижче.

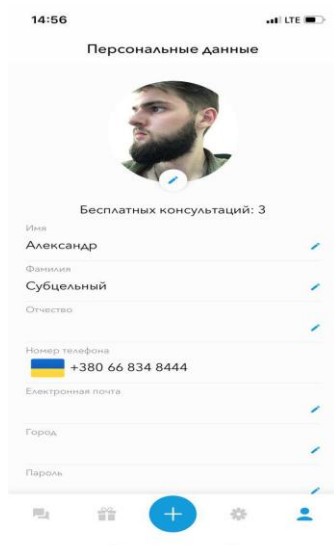


Рисунок 1.6 - Скріншот екрану профіля

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 9 |

1.2.2.2 Змінити профіль

Щоб змінити профіль необхідно:

- а) перейти на вкладку профіля мобільного додатку;
- б) ввести дані форми, а саме, ім'я, прізвище, місто, електронна пошта, пароль. Також можна загрузити фото профіля;
- в) натиснути кнопку “Зберегти”.

1.2.3 Налаштування

Щоб перейти на вкладку налаштування необхідно:

- а) бути авторизованим у системі;
- б) натиснути та перейти на вкладку налаштувань застосунку (малюнок шестерні, другий зправа знизу за стосунку).

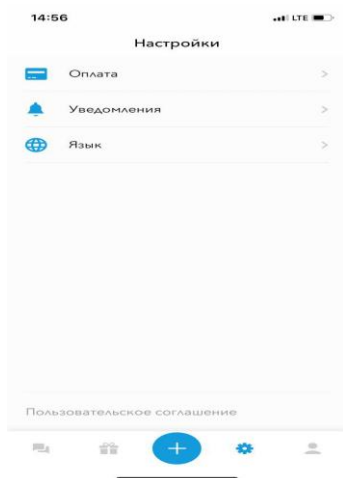


Рисунок 1.7 - Скріншот екрану налаштування

1.2.3.1 Мова

Щоб змінити мову застосунку необхідно:

- а) натиснути вкладку “Мова”;
- б) в нововідкритому вікні обрати мову, натиснувши на плажку з підписом мови.

Дії описані вище наведені на слайдах.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 10 |

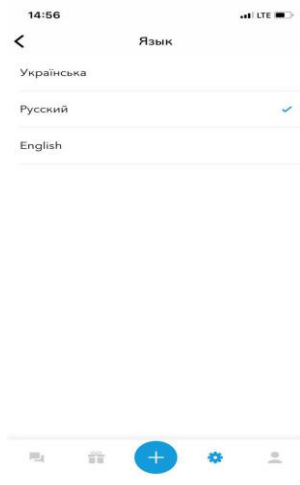


Рисунок 1.8 - Скріншот екрану вибору мови

1.2.3.2 Оповіщення

Щоб змінити налаштування оповіщень потрібно:

- а) натиснути вкладку “Оповіщення”;
- б) в нововідкритому вікні з’явиться 2 плашки для оповіщень нових відповідей від консультантів та системних оповіщень;
- в) перевести плашку з неактивного стану в активний (підсвічений синім кольором).

Дії описані вище наведені на слайдах.

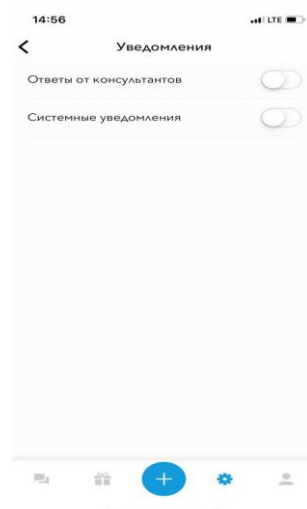
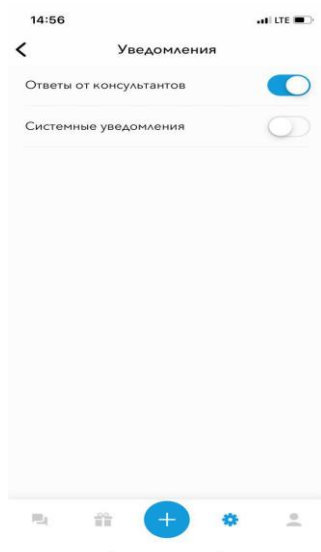


Рисунок 1.9 - Скріншоти екранів оповіщень

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 11 |

1.2.3.3 Оплата

Щоб підключити оплату необхідно:

- а) натиснути вкладку “Оплата”;
- б) в нововідкритому вікні натиснути вкладку “Додати карту”;
- в) в вікні сторінки wayforpay ввести дані своєї карти, а саме, номер карти, місяць на рік закінчення, cvv;
- г) натиснути кнопку “Оплатити”.

Дії описані вище наведені на слайдах.

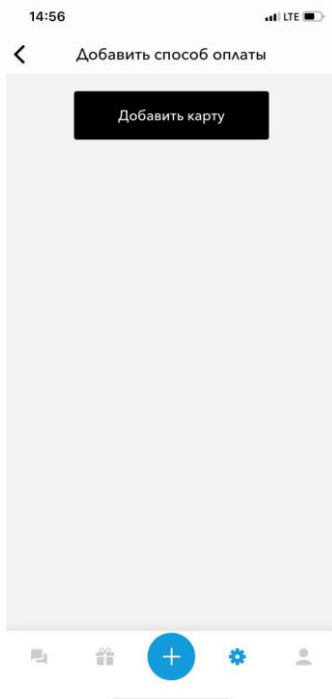


Рисунок 1.10 - Скріншот екрану оплати

1.2.4 Програма лояльності

Щоб прийняти участь у програмі лояльності потрібно:

- а) бути авторизованим в системі;
- б) перейти на вкладку програми лояльності. Значок подарунка, другий зліва знизу додатку;
- в) реферальне посилання буде відображено у спеціальному вікні.

Описані дії наведені на слайдах.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 12 |

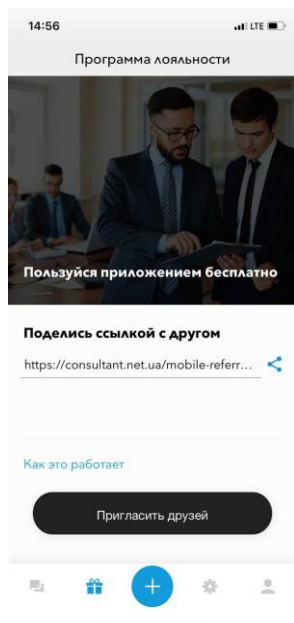


Рисунок 1.11 - Скріншот екрану програми лояльності

1.2.5 Питання

1.2.5.1 Створити питання

Щоб створити питання необхідно:

а) натиснути на “+”(плюсик) посередині плашки знизу мобільного додатку або якщо раніше не було створено питань, на головній сторінці(вона ж “головна”, кнопка зліва на плашці знизу) буде відображатись кнопка “Створити питання”, можна натиснути її;

б) в нововідкритому вікні з'явиться форма для створення питання;

в) заповнити дані форми, а саме: тип питання(кому адресується питання), заголовок та саме питання;

г) натиснути кнопку “Створити”.

Дії описані вище продемонстровані на слайдах.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| | | | | | | 13 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

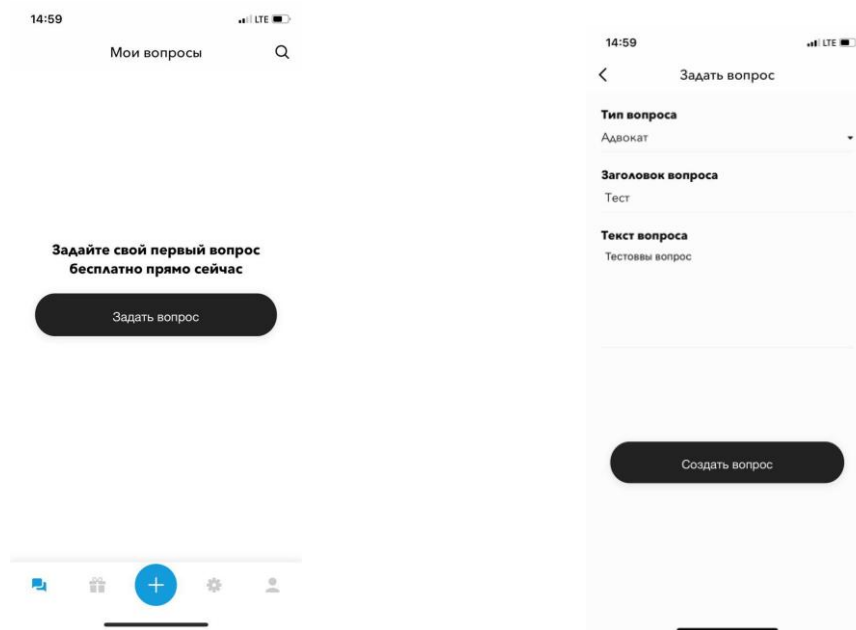


Рисунок 1.12 - Скріншоти екранів створення питання

1.2.5.2 Переглянути список питань

Щоб переглянути всі питання необхідно:

а) натиснути на вкладку головної сторінки(зліва знизу на плашці, значок чату);

б) відобразиться список усіх питань.

Дії описані вище зображені на слайді.

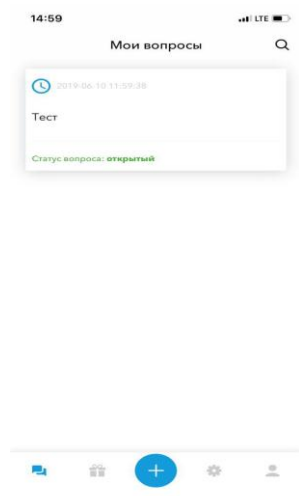


Рисунок 1.13 - Скріншот екрану списку питань

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 14 |

1.2.5.3 Переглянути одне питання

Щоб переглянути питання необхідно:

- а) перейти на головну сторінку(пункт 2.5.2);
- б) обрати питання, що цікавить та натиснути на нього.

Дії описані вище, зображені на слайді.

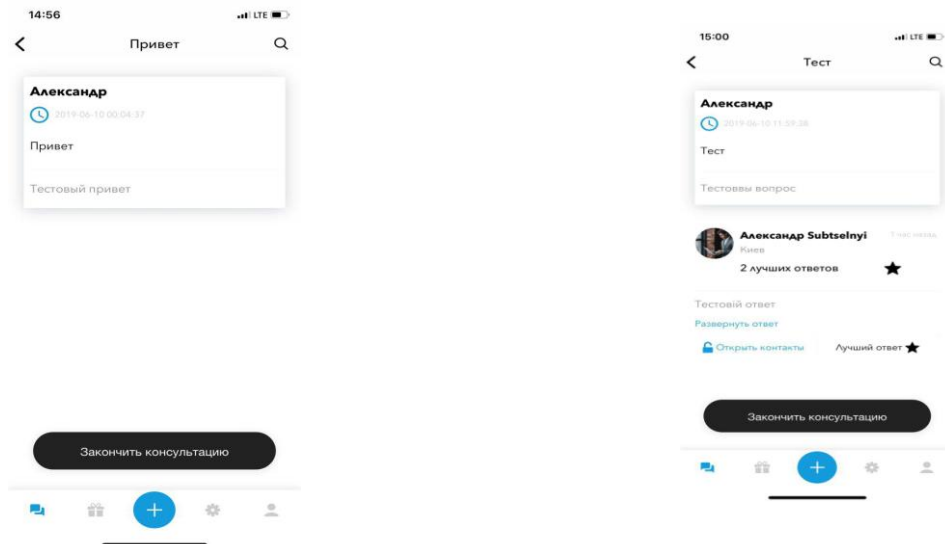


Рисунок 1.14 - Скріншоти екранів перегляду питання

1.2.5.4 Переглянути дані консультанта

Щоб переглянути дані консультанта необхідно:

- а) переглянути одне питання(пункт 2.5.3);
- б) натиснути кнопку “Переглянути дані консультанта”.

Дії описані вище зображені на слайді

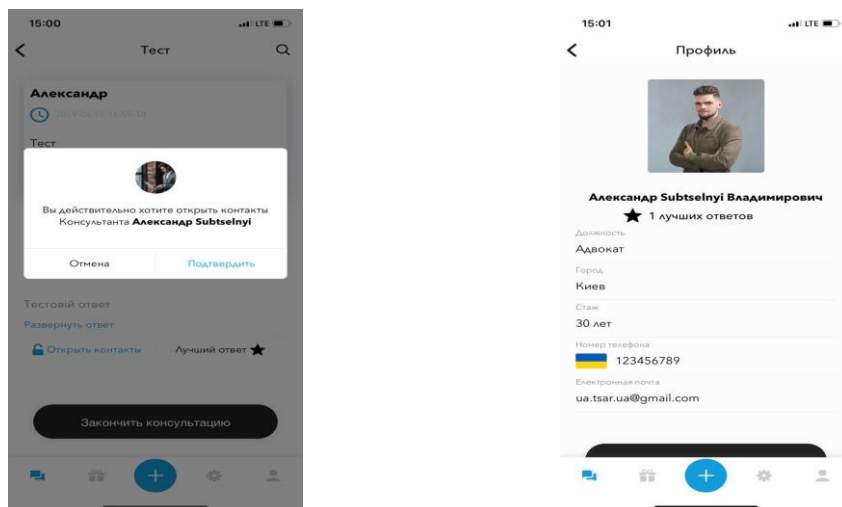


Рисунок 1.15 - Скріншоти екранів перегляду даних консультанта

2 ЧАСТИНА Б

Інструкція Консультанта

Веб-сервіс

2.1 Вимоги до апаратно – програмного забезпечення

Вимоги до апаратно-програмного забезпечення:

а) допустимі наступні версії ОС:

- Windows, Linux, MacOS;

б) додаток адаптований під наступні браузер:

Google Chrome, Mozilla, Safari, Opera;

в) додаток не адаптовано для роботи на планшетах та мобільних телефонах.

2.2 Початок роботи з веб-сервісом

Для роботи з сервісом необхідно вміти працювати з ПК, браузером та мати електронну пошту.

2.2.1 Вхід до Системи

Для входу до сервісу “Консультант”, необхідно відкрити браузер. На екрані з'явиться вікно з описом додатку та кнопкою “Увійти” та “Зареєструватись”.

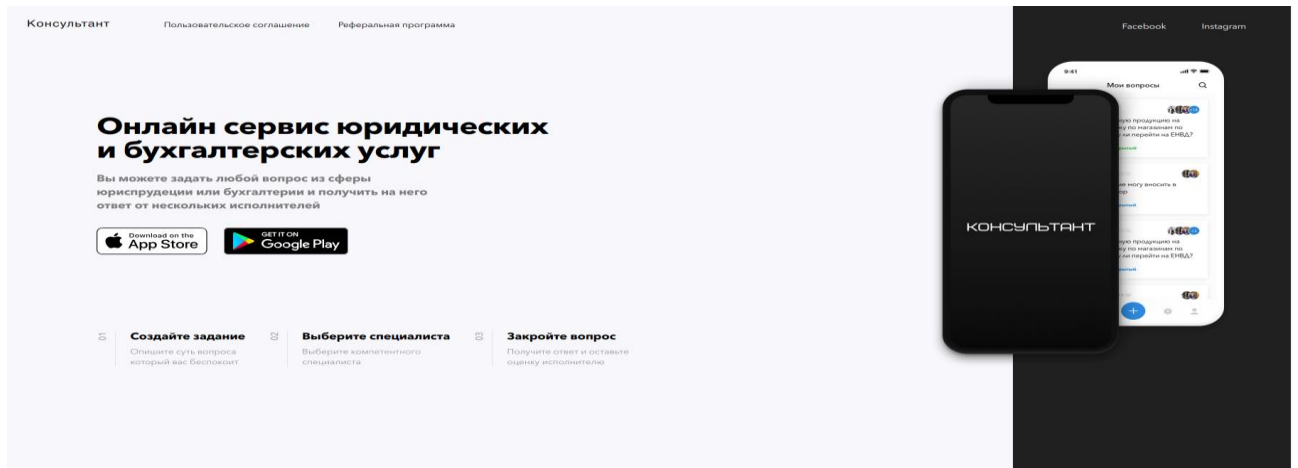


Рисунок 2.1 - Скріншот головного екрану

2.2.1.1 Реєстрація

Щоб зареєструватись необхідно:

- а) на першому вікні обрати надавані послуги;
- б) натиснути кнопку “Далі”;
- в) заповнити дані реєстраційної форми, а саме електронну пошту, ПІБ, номер телефона, пароль на підтвердження пароль, завантажити сертифікат, підтверджуючий статус;
- г) натиснути кнопку “Зареєструватись”;
- д) на пошту буде висланий лист з посиланням-підтвердженням;
- е) необхідно перейти в поштову скриньку, відкрити листа та клацнути на кнопку “Activate”.

Скріншоти описаних дій наведені нижче

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 17 |

Консультант

← Вернуться на главную страницу

Регистрация адвоката

Имя*

Фамилия*

Отчество*

Номер телефона (380XXXXXXXXX)*

E-mail*

Пароль*

Пароль еще раз*

Реферальный код

+

Прикрепить свидетельство

3bc1c1bcced74cd68fa8691f4b3b699a_1024.jpg

Онлайн сервис юридических и бухгалтерских услуг

Facebook

Instagram

Welcome!

You need to activate your email before you can start using all of our services.

Activate

Thank you for using our application!

Regards,
Laravel

If you're having trouble clicking the "Activate" button, copy and paste the URL below into your web browser: https://consulrant.net.ua/activate/zKEGZs1QL8soric43HfxC9vQyJJPWh_Q10oRhaT6kXRh4tZimgCJuvpQmDqE6lcsN

Рисунок 2.2 - Скріншоти екранів реєстрації

2.2.1.2 Авторизація

Щоб авторизуватись необхідно:

- а) натиснути кнопку “Увійти” на головній сторінці;
- б) ввести свою дані в форму, а саме електронну пошту та пароль;
- в) натиснути кнопку “Увійти”

Скріншоти описаних дій наведені нижче

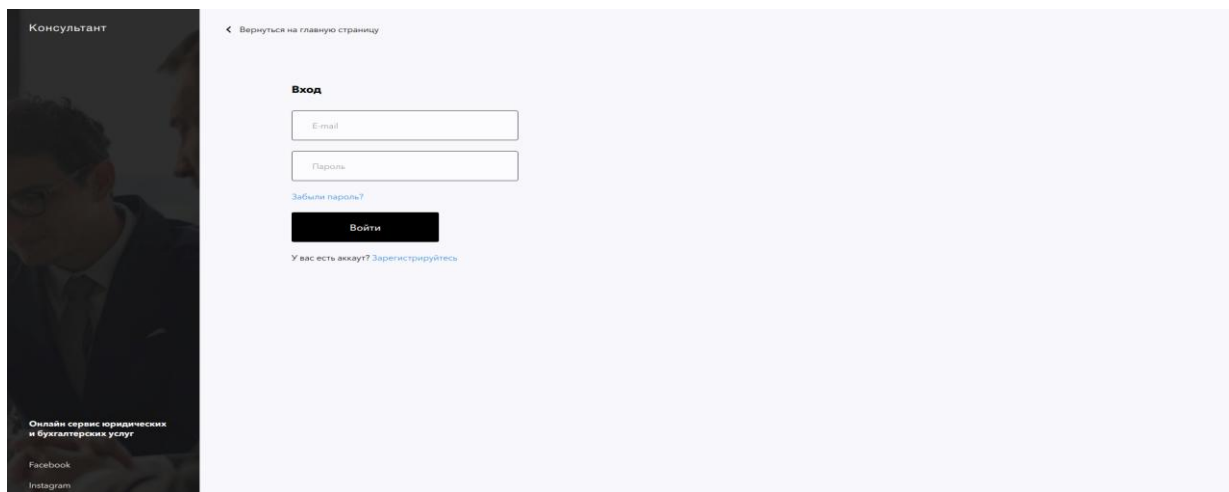


Рисунок 2.3 - Скріншот екрану авторизації

2.2.2 Профіль

2.2.2.1 Переглянути профіль

Щоб переглянути профіль необхідно:

- а) бути авторизованим у веб-версії;
- б) натиснути на аватарку на плашці зверху.

Дії описані вище продемонстровані на слайді нижче.

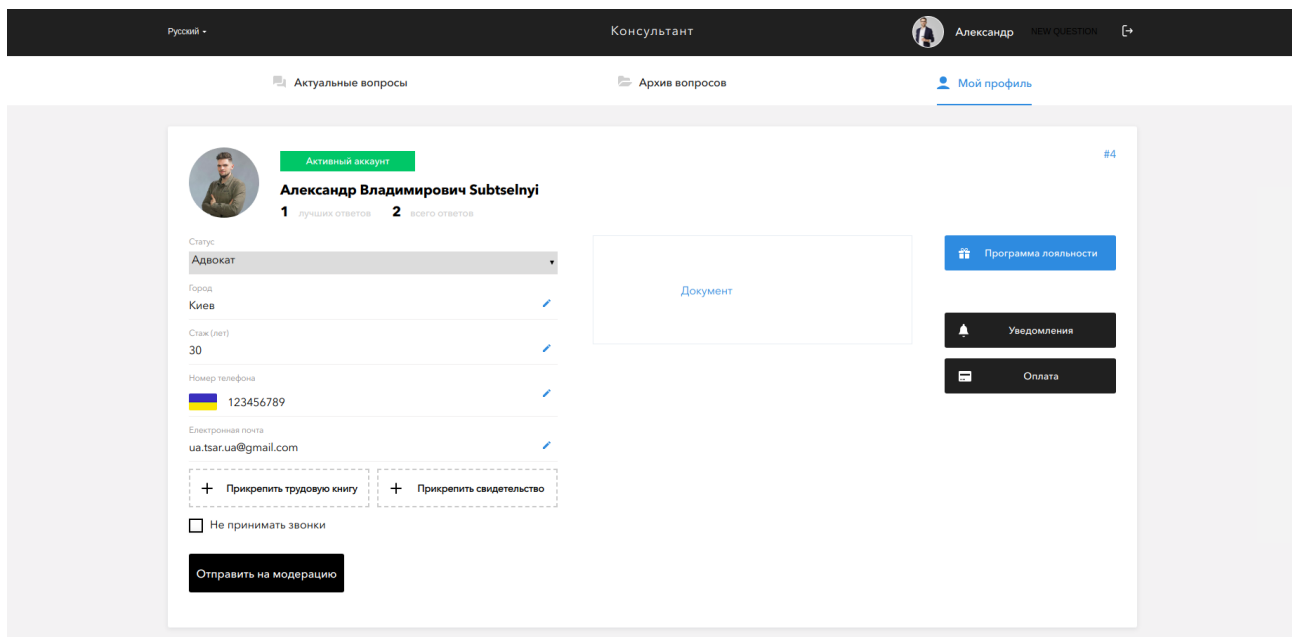


Рисунок 2.4 - Скріншот екрану профіля

2.2.2.2 Змінити профіль

Щоб змінити інформацію профіля необхідно:

- перейти на сторінку профіля(пункт 2.2.1);
- ввести дані, що потрібно змінити у формування та завантажити документи;
- натиснути кнопку “Відправити модератору”;
- дані профіля будуть відправлені в адміністративну панель, де будуть провалідовані модератором, після чого або збережені, а бо відхилені в автоматичному режимі.

2.2.3 Налаштування

2.2.3.1 Програма лояльності

Щоб прийняти участь у програмі лояльності необхідно:

- перейти на сторінку профіля(пункт 2.2.1);
 - натиснути кнопку “Програма лояльності”;
 - поділитися своєю реферальною ссилкою, що відображення у формі.
- Слайди описаних дій наведені нижче.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 20 |

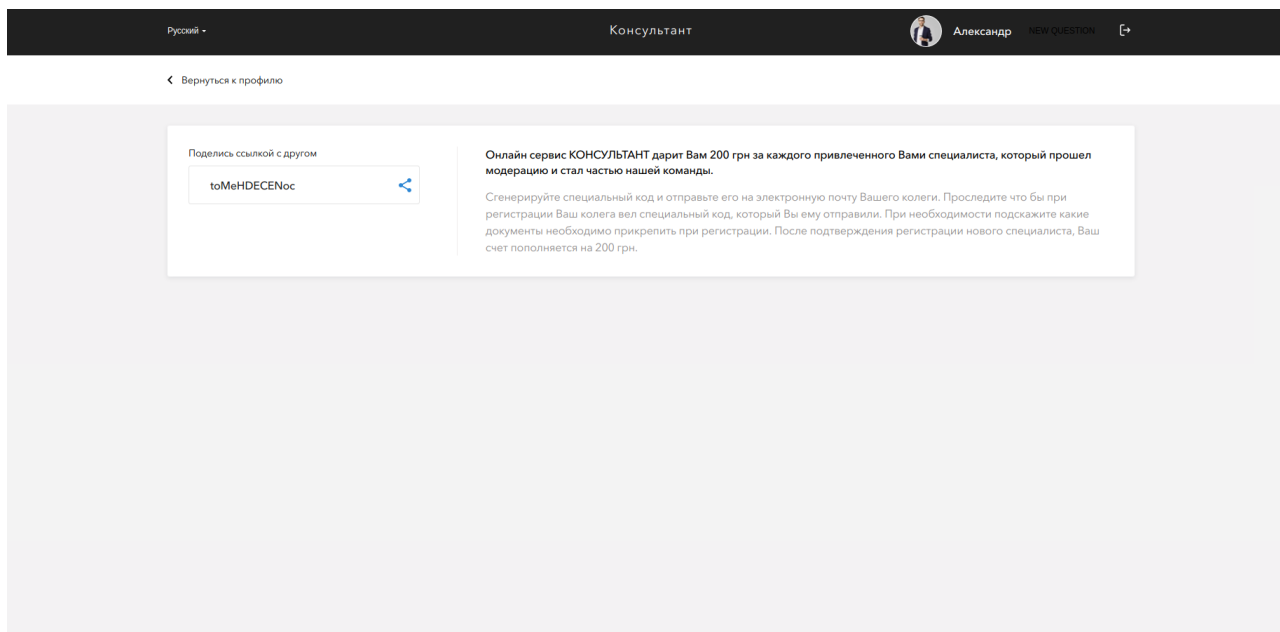


Рисунок 2.5 - Скріншот екрану програми лояльності

2.2.3.2 Оповіщення

Щоб налаштувати оповіщення необхідно:

- а) перейти на сторінку профіля(пункт 2.2.1);
- б) натиснути кнопку “Оповіщення”;
- в) перемістити повзунки плашки вправо або вліво(вімк./вимк.), якщо повзунок горить синім – функція ввімкнена;
- г) Натиснути кнопку “Зберегти”.

Слайди описаних дій наведені нижче.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 21 |

[← Вернуться к профилю](#)

Вопросы от клиентов

☐

Системные уведомления

☐

Сохранить

[← Вернуться к профилю](#)

Вопросы от клиентов

☒

Системные уведомления

☐

Сохранить

Рисунок 2.6 - Скріншоти екранів оповіщення

2.2.3.3 Оплата

Щоб підключити оплату необхідно:

- а) перейти на сторінку профіля(пункт 2.2.1);
- б) натиснути кнопку “Оплата”;
- в) у нововідкритому вікні натиснути кнопку “Додати картку”;
- г) у вікні сайту wayforpay ввести дані своєї картки, а саме номер картки, місяць на рік закінчення строку дії, cvv-код;
- д) натиснути кнопку “Оплатити”.

Дії описані вище наведені на слайдах.

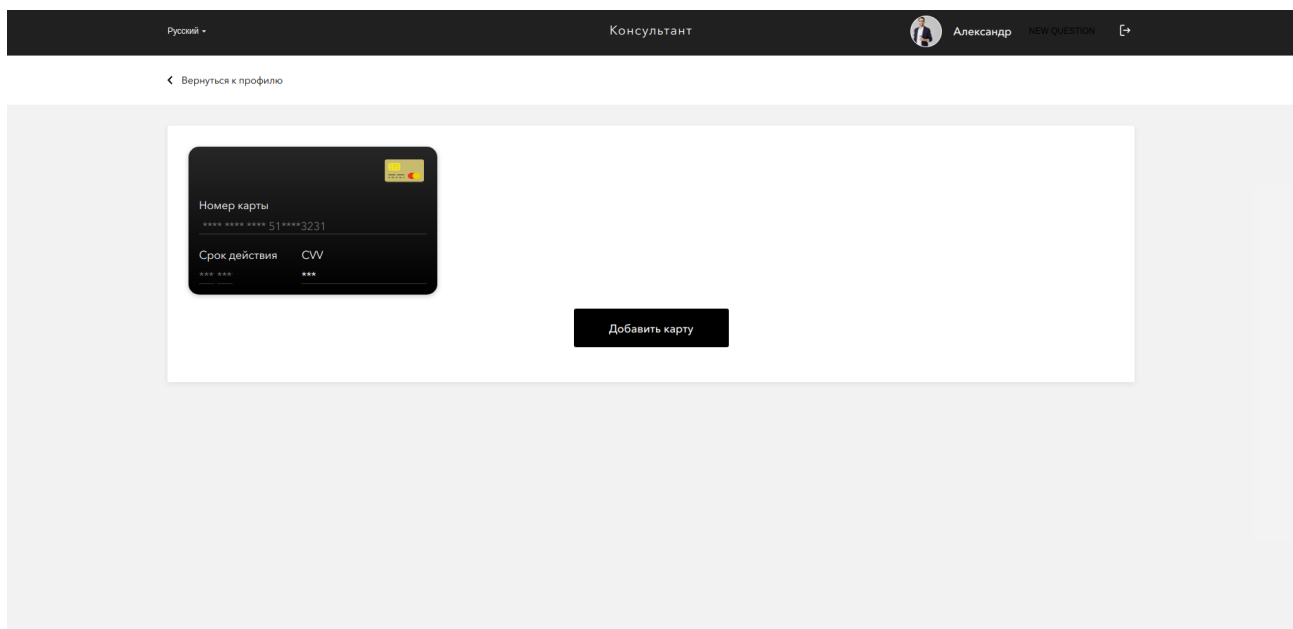


Рисунок 2.7 - Скріншот екрану оплати

2.2.4 Питання – Відповіді

Щоб перейти на список питань клієнтів необхідно:

- а) натиснути вкладку “Активні питання” зверху на плашці

2.2.4.1 Відповісти на запитання

Щоб відповісти на питання необхідно:

- а) перейти на сторінку питань(пункт 2.4);
- б) обрати потрібне питання та клацнути на нього;
- в) у відкритому діалоговому вікні ввести текст відповіді;

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 23 |

- г) натиснути кнопку “Надіслати”;
- д) діалогове вікно закриється, а саме питання пропаде зі сторінки питань, та зявиться у вкладці “Архів питань”.

Дії описані вище наведені на слайдах.

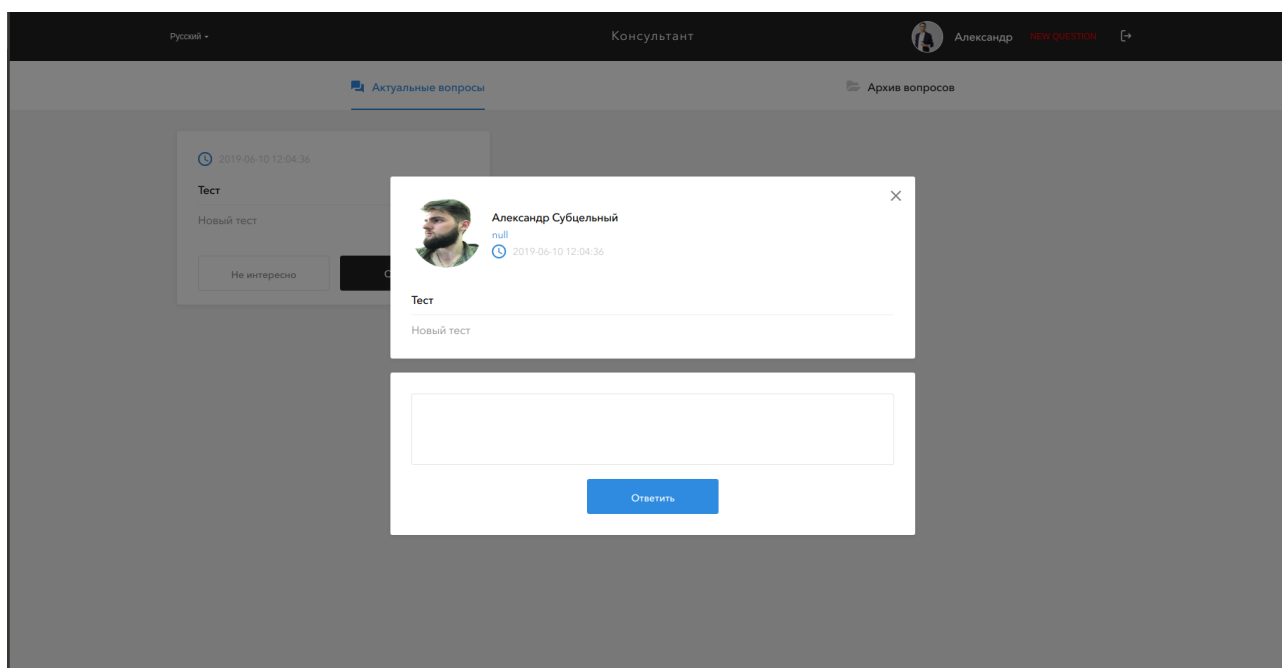
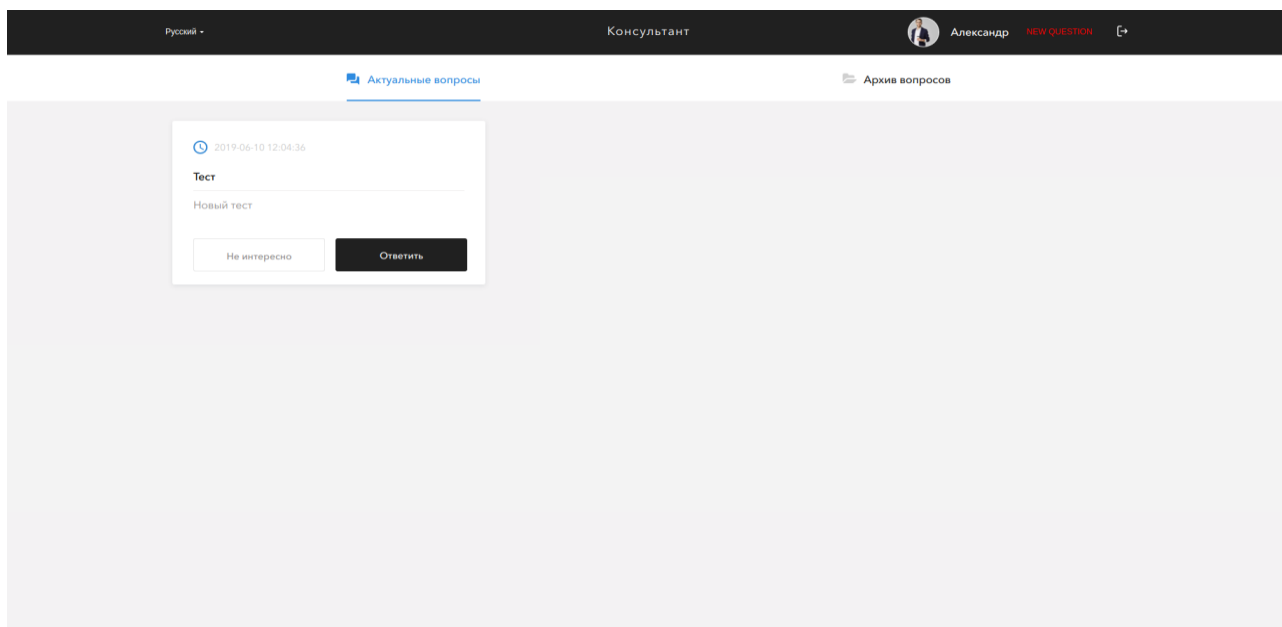


Рисунок 2.8 - Скріншоти екрану питання - відповідь

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 24 |

2.2.4.2 Переглянути відповіді

Щоб переглянути відповіді необхідно:

а) натиснути на вкладку “Архів питань” на плашці зверху.

Слайди наведені нижче.

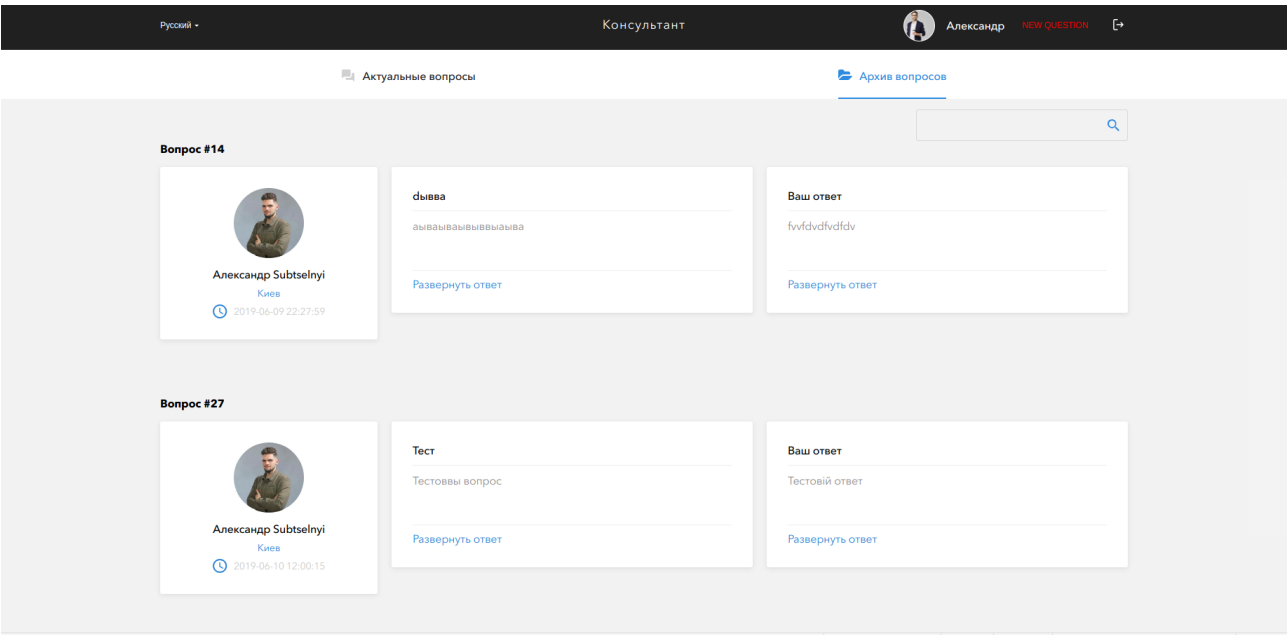


Рисунок 2.9 - Скріншот екрану архіву питань

3 ЧАСТИНА В

Інструкція Модератора Веб-сервіс

3.1 Вимоги до апаратно – програмного забезпечення

Вимоги до апаратно-програмного забезпечення:

а) допустимі наступні версії ОС:

Windows, Linux, MacOS;

б) додаток адаптований під наступні браузери:

Google Chrome, Mozilla, Safari, Opera;

в) додаток не адаптовано для роботи на планшетах та мобільних телефонах.

3.2 Початок роботи з веб-сервісом

Для роботи з сервісом необхідно вміти працювати з ПК, браузерам та мати електронну пошту.

3.2.1 Вхід до Системи

Для входу в адміністративну панель необхідно вводити спеціальне посилання в строку браузера /adminlogin. Веб-сервіс перенаправить користувача на таємну сторінку авторизації.

3.2.1.1 Авторизація

Щоб авторизуватись як модератор необхідно:

а) ввести на сторінці авторизації дані форми, а саме електронну пошту та пароль;

б) натиснути кнопку “Увійти”.

Слайди наведені нижче.

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | КПІ.ІП-5121.045440.05.34 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 26 |

Laravel Login Register

Login

E-Mail Address

Password

[Login](#)

Рисунок 3.1 - Скріншот екрану авторизації

3.2.2 Керування користувачами

3.2.2.1 Перегляд користувачів

Щоб переглянути усіх користувачів, достатньо бути авторизованим як модератор. Список усіх користувачів висвітлюється на головній сторінці.

Слайди наведені нижче

Laravel Admin

Showing All Users

| ID | Email | First Name | Last Name | Role | Type | Created | Updated | Actions |
|----|------------------|------------|------------|------------|---------|---------------------|---------------------|---|
| 2 | admin@admin.com | admin | admin | User | Клієнт | 2019-06-03 13:43:16 | 2019-06-03 13:43:16 | Delete User |
| 3 | ua.ua@gmail.com | Ов | Ос | Unverified | Клієнт | 2019-06-03 14:48:25 | 2019-06-07 13:30:47 | Delete User |
| 4 | ua.ua@gmail.com | Александр | Subtselnyi | Consultant | Адвокат | 2019-06-04 13:18:05 | 2019-06-10 12:03:14 | Delete User Show User Edit User |
| 5 | test@test.com | Александр | Субцельный | Unverified | Клієнт | 2019-06-04 13:41:30 | 2019-06-10 11:56:17 | Delete User Show User Edit User |
| 6 | test@test.com | Дима | Лузанов | Unverified | Клієнт | 2019-06-06 12:58:20 | 2019-06-10 10:57:00 | Delete User Show User Edit User |
| 7 | test@test.com | Лавовал | Чтазово | Unverified | Клієнт | 2019-06-06 17:44:58 | 2019-06-06 17:44:58 | Delete User Show User Edit User |
| 8 | test@test.com | Алекс | Слит | Unverified | Клієнт | 2019-06-07 08:32:45 | 2019-06-10 10:53:10 | Delete User Show User Edit User |
| 9 | test@test.com | Тату | Вов | Unverified | Клієнт | 2019-06-09 09:25:18 | 2019-06-09 09:25:18 | Delete User Show User Edit User |
| 10 | dimak.86@ukr.net | Dima | Luzanov | Consultant | Юрист | 2019-06-09 12:32:22 | 2019-06-09 13:18:49 | Delete User Show User Edit User |
| 11 | test@test.com | Dima | Luzanov | Unverified | Клієнт | 2019-06-09 21:43:55 | 2019-06-09 21:50:18 | Delete User Show User Edit User |
| 12 | juzya@gmail.com | ДИМА | ДИМАСИ | Consultant | Юрист | 2019-06-10 09:32:13 | 2019-06-10 09:38:15 | Delete User Show User Edit User |

12 user total

- [New User](#)
- [Show Deleted User](#)
- [Клієнти](#)
- [Не подтвержденные консультанты](#)
- [Консультанты](#)
- [Консультанты на модерации](#)

Рисунок 3.2 - Скріншот екрану перегляду користувачів

3.2.2.2 Переглянути користувача

Щоб переглянути користувача необхідно:

- а) переглянути усіх користувачів(пункт 2.2.1);
- б) обрати користувача та натиснути кнопку “Show User”.

Слайд продемонстрований нижче.

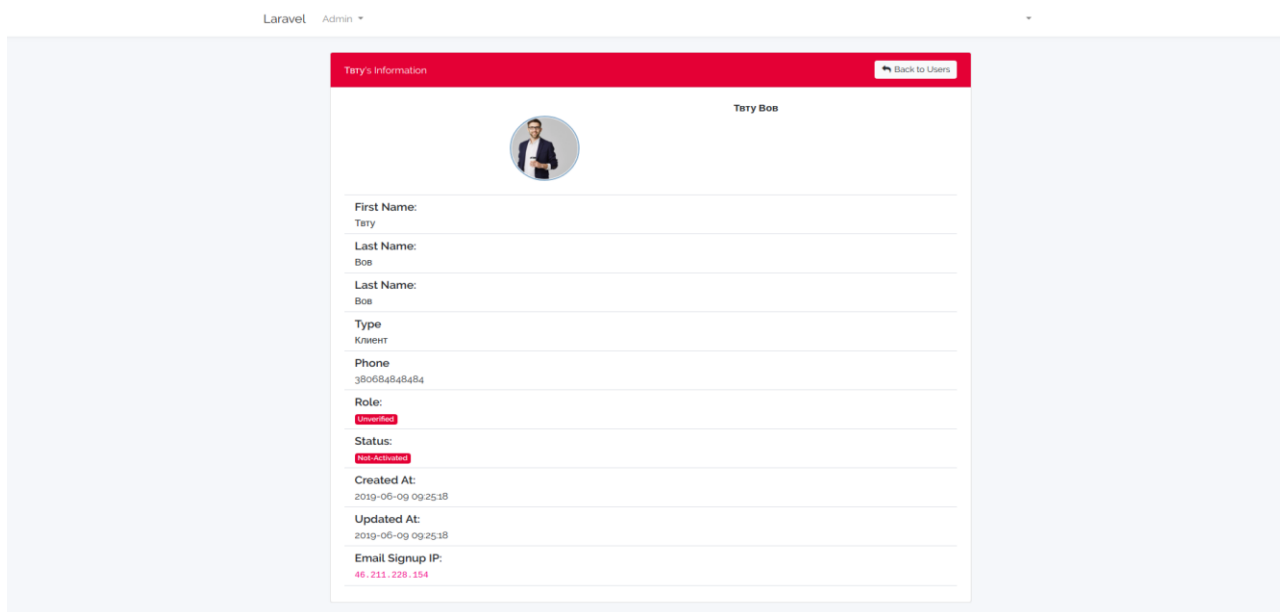


Рисунок 3.3 - Скріншот екрану перегляду користувача

3.2.2.3 Верифікувати консультанта/ змінити профіль користувача

Щоб Верифікувати консультанта або змінити профіль користувача необхідно:

- а) переглянути усіх користувачів (пункт 3.2.2.1);
- б) обрати користувача та натиснути кнопку “Edit User”.

Слайд наведений нижче.

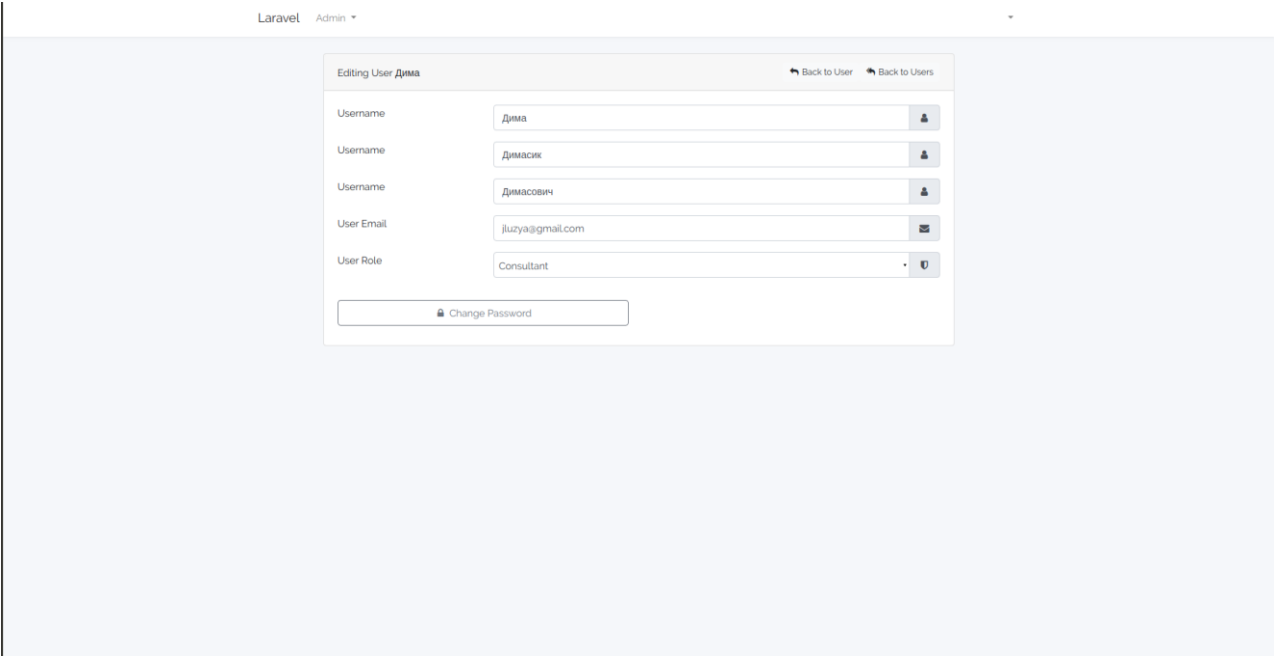


Рисунок 3.4 - Скріншот екрану верифікації консультанта

4 ЧАСТИНА Г

Інструкція Адміністратора Веб-сервіс

4.1 Вимоги до апаратно – програмного забезпечення

Вимоги до апаратно-програмного забезпечення:

а) допустимі наступні версії ОС:

Windows, Linux, MacOS;

б) додаток адаптований під наступні браузери:

Google Chrome, Mozilla, Safari, Opera;

в) додаток не адаптовано для роботи на планшетах та мобільних телефонах.

4.2 Початок роботи з веб-сервісом

Для роботи з сервісом необхідно вміти працювати з ПК, браузером та мати електронну пошту.

4.2.1 Вхід до Системи

Для входу в адміністративну панель необхідно вводити спеціальне посилання в строку браузера /adminlogin. Веб-сервіс перенаправить користувача на таємну сторінку авторизації.

4.2.1.1 Авторизація

Щоб авторизуватись як адміністратор необхідно:

а) ввести на сторінці авторизації дані форми, а саме електронну пошту та пароль;

б) натиснути кнопку “Увійти”.

Слайди наведені нижче.

| | | | | | | |
|------|------|----------|--------|------|-------------------------|------|
| | | | | | КП.ІП-5121.045440.05.34 | Арк. |
| | | | | | | 30 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Laravel Login Register

Login

E-Mail Address

Password

[Login](#)

Рисунок 4.1 - Скрішот екрану авторизації

4.2.2 Керування користувачами

4.2.2.1 Перегляд користувачів

Щоб переглянути усіх користувачів, достатньо бути авторизованим як адміністратор. Список усіх користувачів висвітлюється на головній сторінці.

Слайди наведені нижче.

Laravel Admin

Showing All Users

| ID | Email | First Name | Last Name | Role | Type | Created | Updated | Actions |
|----|----------------------|------------|------------|----------------------|---------|---------------------|---------------------|---|
| 2 | admin@admin.com | admin | admin | View | Клієнт | 2019-06-03 13:43:16 | 2019-06-03 13:43:16 | Delete User |
| 3 | | Ов | Ос | View | Клієнт | 2019-06-03 14:48:25 | 2019-06-07 13:30:47 | Delete User |
| 4 | ua.tsar.ua@gmail.com | Александр | Subtseinyi | View | Адвокат | 2019-06-04 13:18:05 | 2019-06-10 12:03:14 | Delete User Show User Edit User |
| 5 | | Александр | Субцельный | View | Клієнт | 2019-06-04 13:41:30 | 2019-06-10 11:56:17 | Delete User Show User Edit User |
| 6 | | Діма | Лузанов | View | Клієнт | 2019-06-06 12:58:20 | 2019-06-10 10:57:00 | Delete User Show User Edit User |
| 7 | | Лавоовал | Чтаовоов | View | Клієнт | 2019-06-06 17:44:58 | 2019-06-06 17:44:58 | Delete User Show User Edit User |
| 8 | test@test.com | Алекс | Смит | View | Клієнт | 2019-06-07 08:32:45 | 2019-06-10 10:53:10 | Delete User Show User Edit User |
| 9 | | Tetu | Вов | View | Клієнт | 2019-06-09 09:25:18 | 2019-06-09 09:25:18 | Delete User Show User Edit User |
| 10 | dimak.86@ukr.net | Dima | Luzaanov | View | Юрист | 2019-06-09 12:32:22 | 2019-06-09 13:18:49 | Delete User Show User Edit User |
| 11 | | Dima | Luzaanov | View | Клієнт | 2019-06-09 21:43:55 | 2019-06-09 21:56:18 | Delete User Show User Edit User |
| 12 | jlucyad@gmail.com | Діма | Димасик | View | Юрист | 2019-06-10 09:32:13 | 2019-06-10 09:38:15 | Delete User Show User Edit User |

12 user total

[New User](#)
[Show Deleted User](#)
[Клієнти](#)
[Не подтвержденные консультанты](#)
[Консультанты](#)
[Консультанты на модерации](#)

Рисунок 4.2 - Скріншот екрану перегляду користувачів

4.2.2.2 Переглянути користувача

Щоб переглянути користувача необхідно:

- а) переглянути усіх користувачів(пункт 2.2.1);
- б) обрати користувача та натиснути кнопку “Show User”.

Слайд продемонстрований нижче.

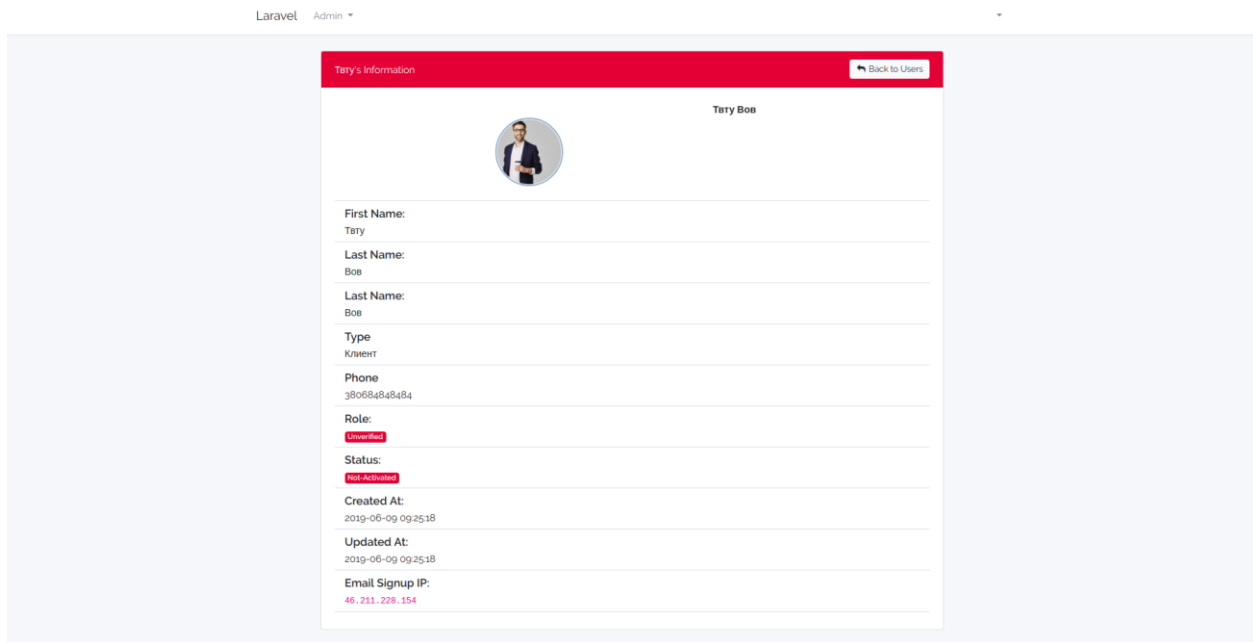


Рисунок 4.3 - Скріншот екрану перегляду користувача

4.2.2.3 Верифікувати консультанта/ змінити профіль користувача

Щоб Верифікувати консультанта або змінити профіль користувача необхідно:

- а) переглянути усіх користувачів(пункт 2.2.1);
- б) обрати користувача та натиснути кнопку “Edit User”.

Слайд наведений нижче.

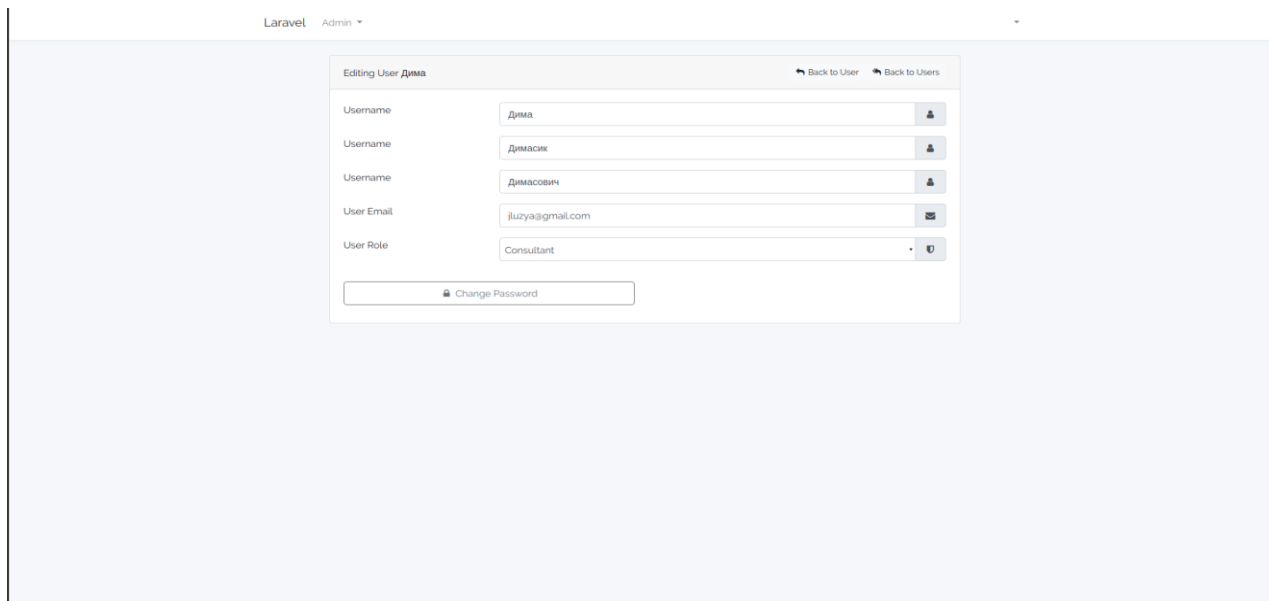


Рисунок 4.4 - Скріншот екрану верифікації консультанта

4.2.2.4 Створити нового користувача

Щоб створити нового користувача необхідно:

- а) бути авторизованим як АДМІНІСТРАТОР;
- б) на верхній плашці обрати опцію “Create User”;
- в) ввести дані форми на сторінці створення користувачам;
- г) натиснути кнопку “Save”.

Слайд наведений нижче.

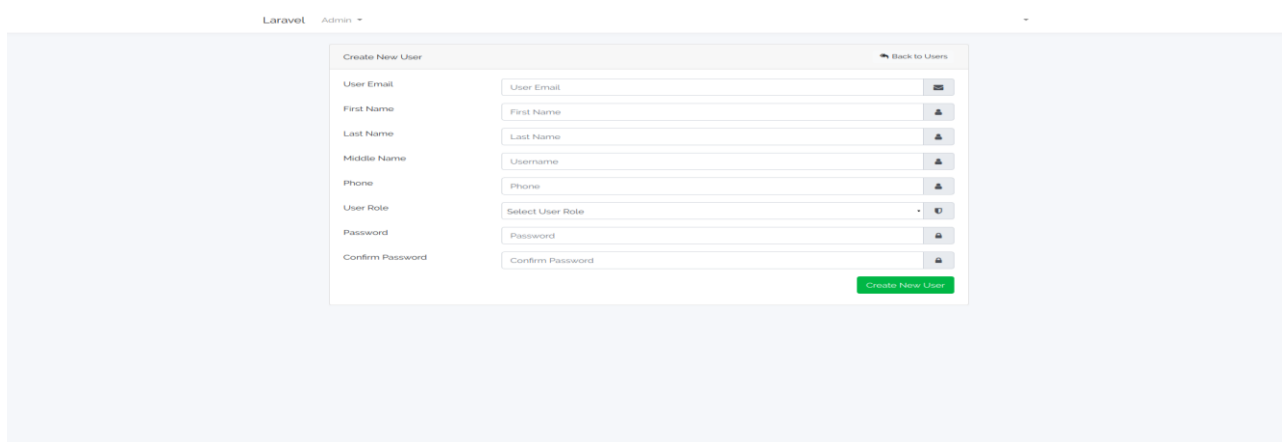
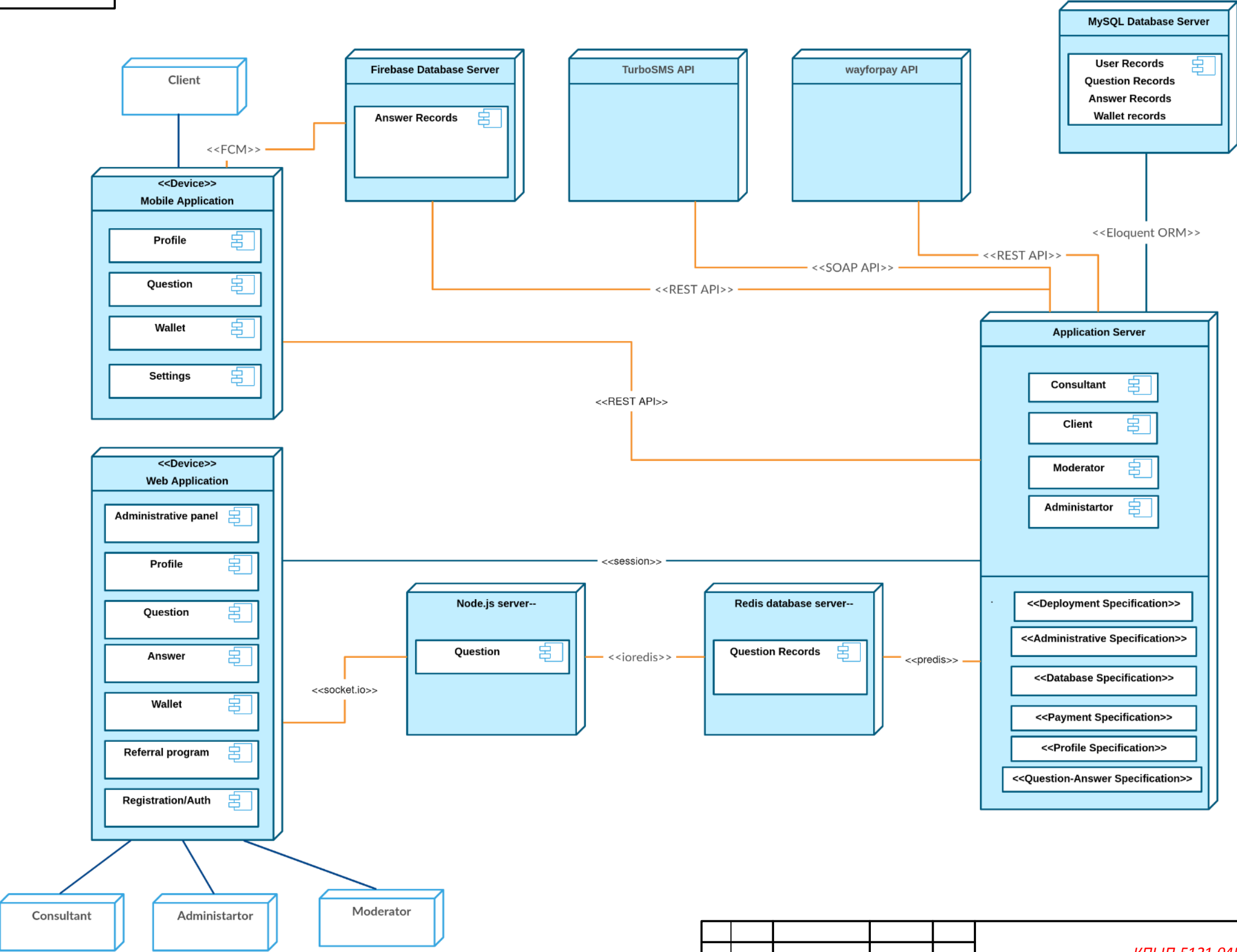
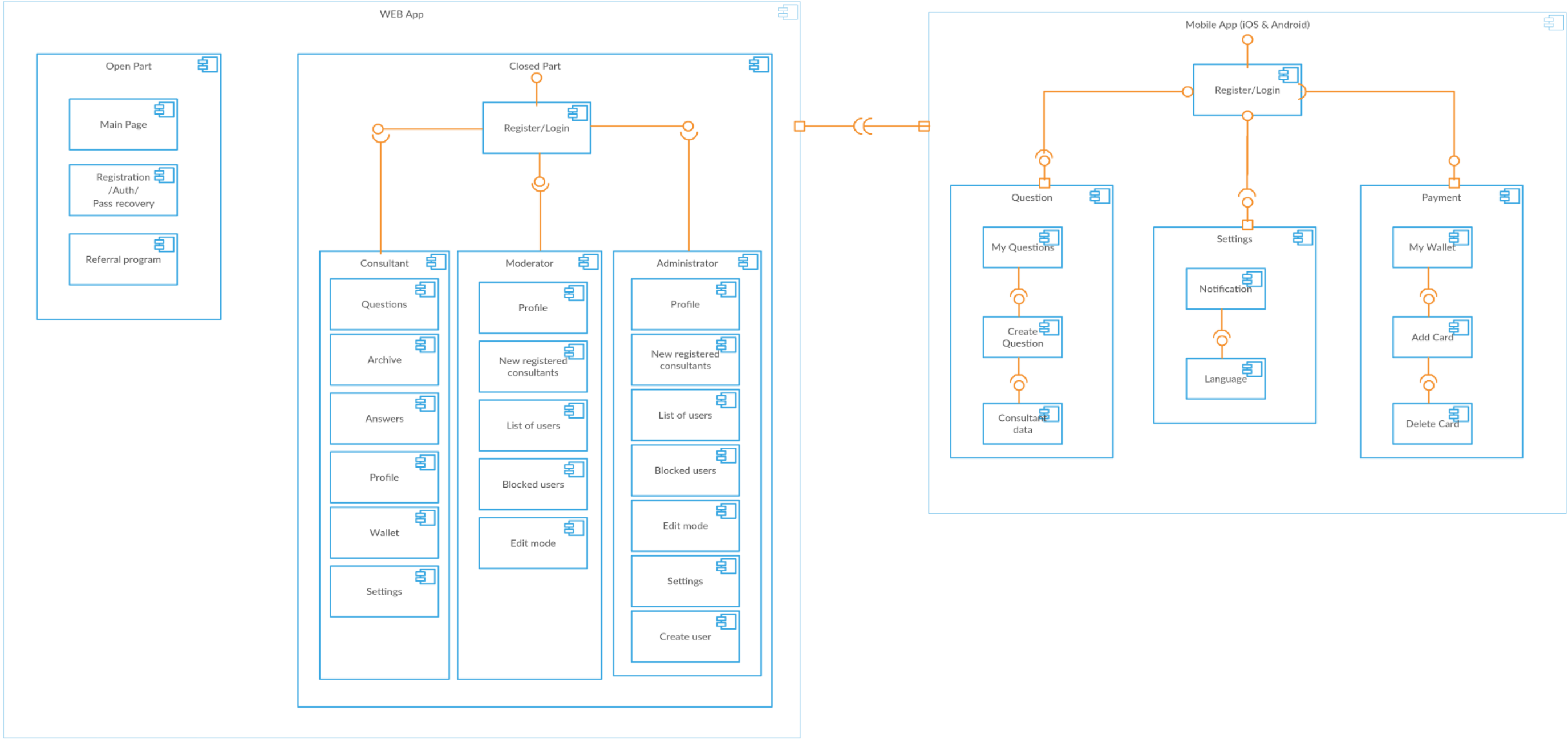


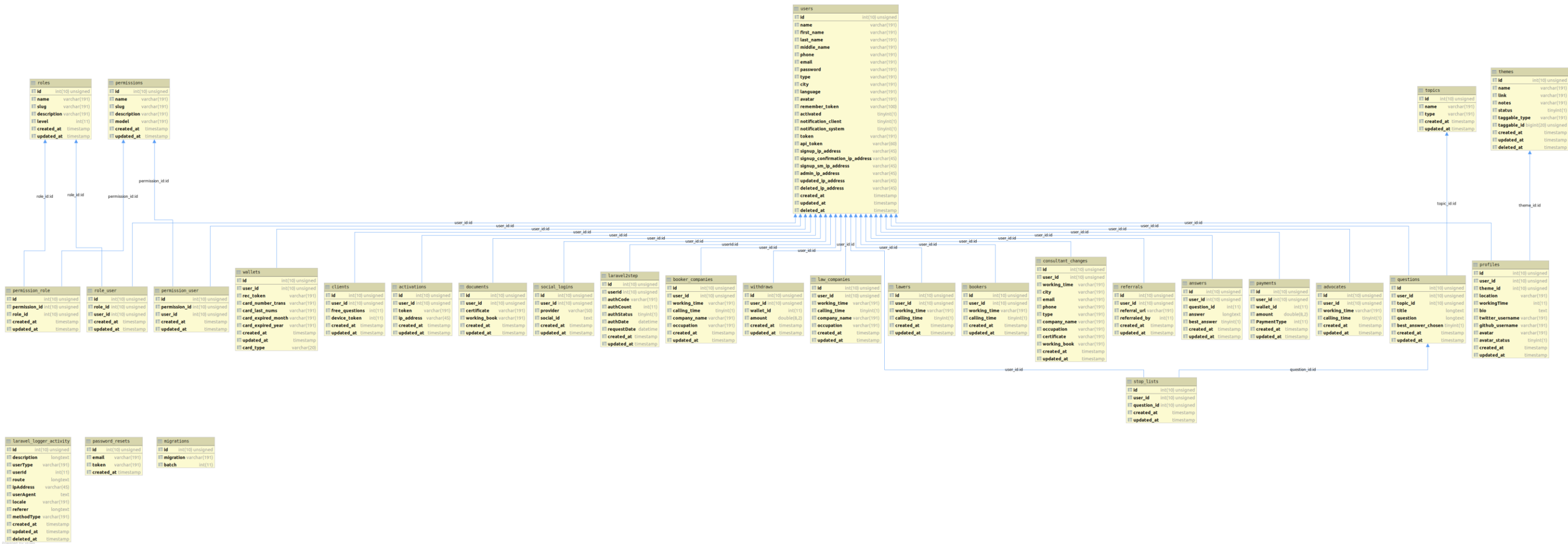
Рисунок 4.5 - Скріншот екрану створення нового користувача



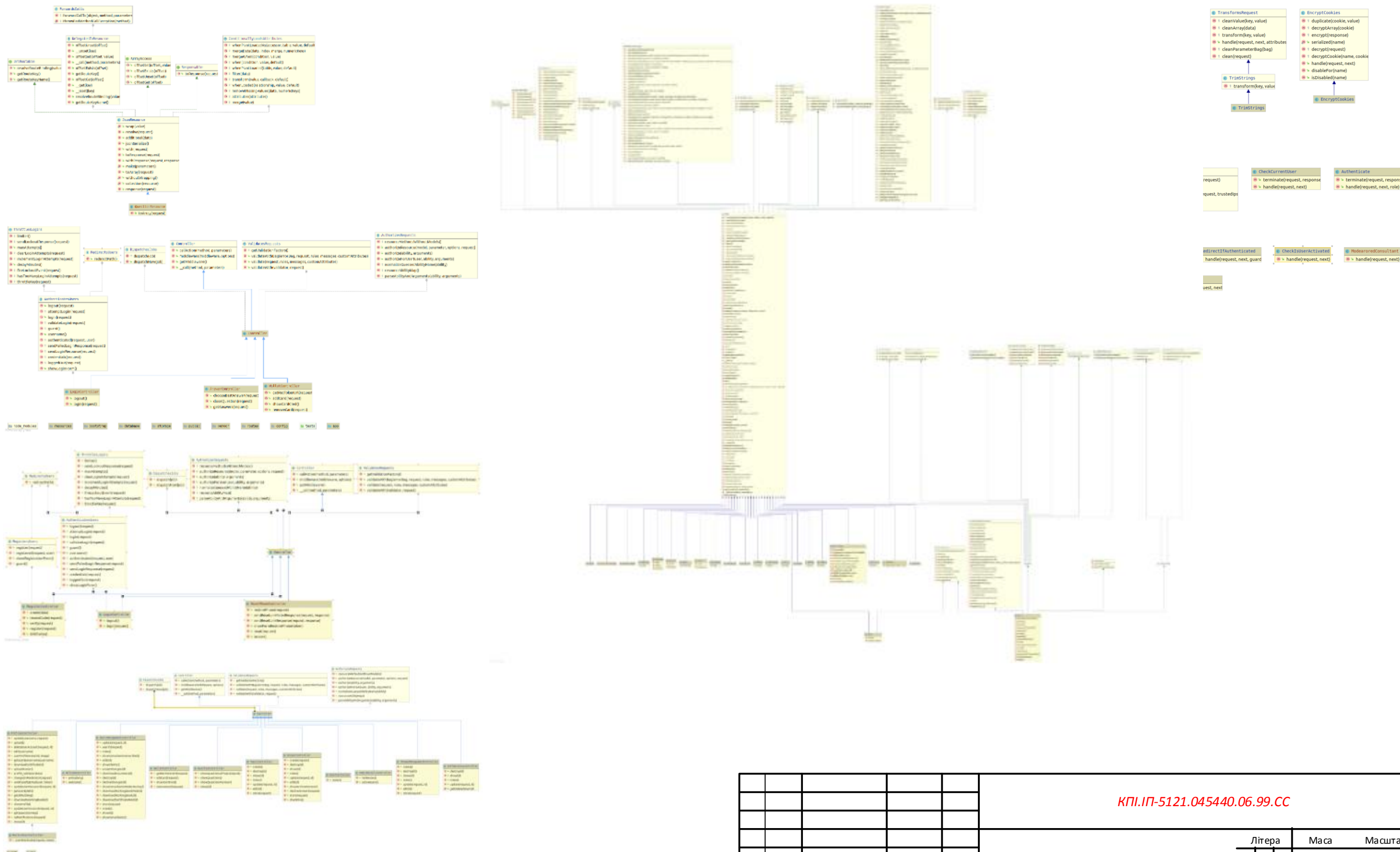
| | | | | | | | | |
|-----------|------|-----------------|--------|------|---|--|---------|---------|
| | | | | | | КПІ.ІП-5121.045440.06.99.СС | | |
| | | | | | | | | |
| | | | | | | | | |
| Зм. | Арк. | № документа | Підпис | Дата | Схема структурна розгортання | Літера | Маса | Масштаб |
| Розробив | | Субцельний О.В. | | | | | | |
| Перевірив | | Муха І.П. | | | | | | |
| Т. кон. | | | | | | Аркуш | Аркушів | |
| Н. кон. | | Ліщук К.І. | | | Онлайн-платформа для надання юридичних послуг | КПІ ім.Ігоря СікорськогоКафедра АСОІУгр. ІП-51 | | |
| Затвердив | | Муха І.П. | | | | | | |



| | | | | | | | | | |
|-----------|------|-----------------|--------|------|---|---|--|---------|---------|
| | | | | | КПІ.ІП-5121.045440.06.99.СС | | | | |
| | | | | | Схема структурна компонентів програмного забезпечення | Літера | | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | | | |
| Розробив | | Субцельний О.В. | | | | | | | |
| Перевірів | | Муха І.П. | | | | | | | |
| Т. кон. | | | | | | Аркуш | | Аркушів | |
| Н. кон. | | Ліщук К.І. | | | Онлайн-платформа для надання юридичних послуг | КПІ ім.Ігоря СікорськогоКафедра АСОІур. ІП-51 | | | |
| Затвердив | | Муха І.П. | | | | | | | |

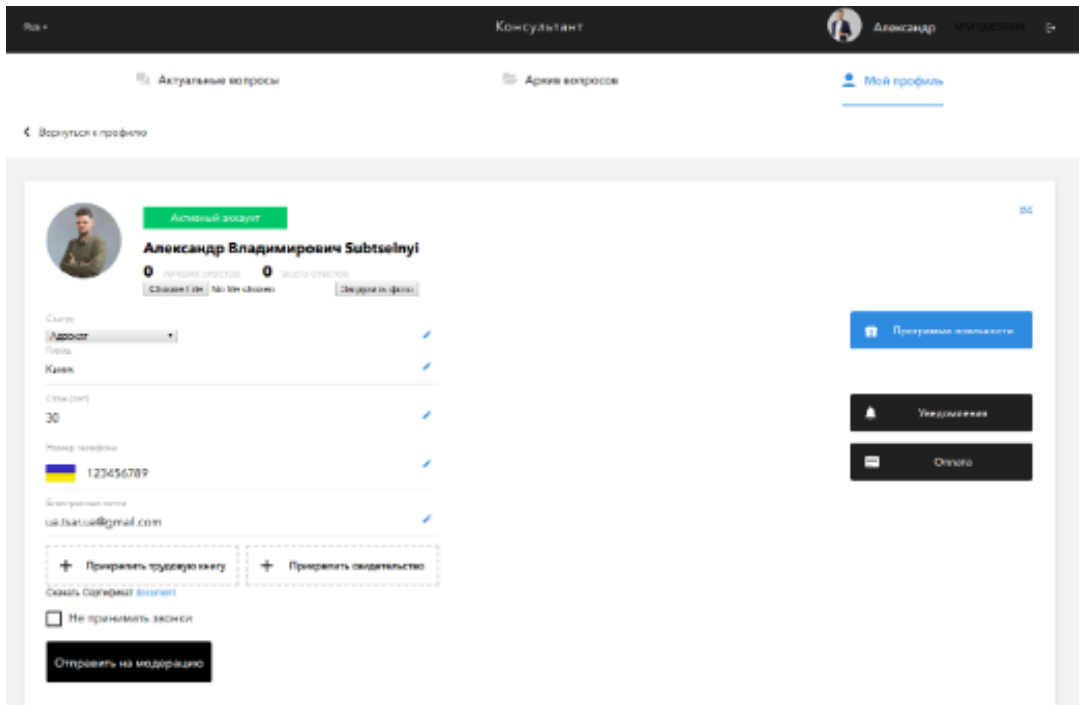
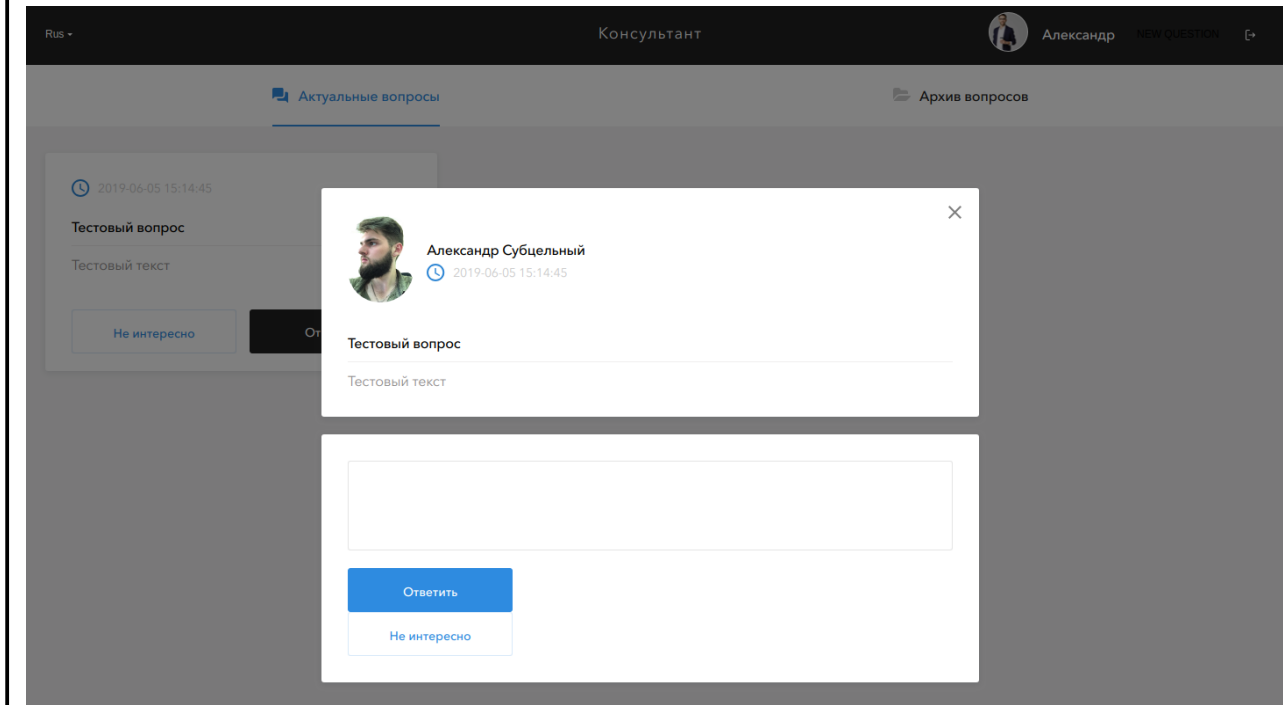


| | | | | | |
|-----------|------|-----------------|--------|------|---|
| | | | | | КП.ІП-5121.045440.06.99.СБД |
| | | | | | |
| | | | | | |
| | | | | | |
| Зм. | Арк. | № документа | Підпис | Дата | Схема бази даних |
| Розробив | | Субцельний О.В. | | | |
| Перевірив | | Муха І.П. | | | |
| Т. кон. | | | | | |
| Н. кон. | | Ліщук К.І. | | | Онлайн-платформа для надання юридичних послуг |
| Затвердив | | Муха І.П. | | | |
| | | | | | Літера |
| | | | | | Маса |
| | | | | | Масштаб |
| | | | | | Аркуш |
| | | | | | Аркушів |
| | | | | | КП ім.Ігоря СікорськогоКафедра АСОІУпр. ІП-51 |



| | | | | | | | | | |
|--|--|--|--|--|--|-----------------------------|--|--|--|
| | | | | | | КПІ.ІП-5121.045440.06.99.СС | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

КП.ІП-5121.045440.06.99.КЕ



18:18

Добавить способ оплаты

Название товара: cardVerification

Всего: 1 UAH

VISA masterpass

Form for adding a payment method. It includes fields for: Номер карты (XXXX XXXX XXXX XXXX), Срок действия (MM/YY), Имя на карте, Код CVV2 (XXX), Email (example@gmail.com), and Номер телефона (+380). There is a blue button labeled 'Оплатить'.

Laravel Admin


Showing All Users

| ID | Email | First Name | Last Name | Role | Type | Created | Updated | Actions |
|----|----------------------|------------|------------|--------|--------|---------------------|---------------------|---|
| 2 | admin@admin.com | admin | admin | Admin | Client | 2019-05-24 14:47:35 | 2019-05-24 14:47:35 | View User Edit User Delete User |
| 10 | ua.tour.ua@gmail.com | Александр | Субцельный | client | Admin | 2019-05-24 15:00:33 | 2019-05-28 07:57:00 | View User Edit User Delete User |
| 16 | | Александр | Субцельный | client | Client | 2019-05-24 15:00:34 | 2019-05-24 15:03:03 | View User Edit User Delete User |
| 17 | dena | Людмила | Людмила | client | Client | 2019-05-24 15:02:08 | 2019-05-24 15:23:37 | View User Edit User Delete User |
| 18 | test@ag.com | Alex | Smith | client | Client | 2019-05-24 15:30:00 | 2019-05-27 20:09:59 | View User Edit User Delete User |
| 19 | | Alex | Smith | client | Client | 2019-05-24 15:34:18 | 2019-05-24 15:34:18 | View User Edit User Delete User |
| 20 | | Alex | Smith | client | Client | 2019-05-24 15:35:08 | 2019-05-24 15:35:08 | View User Edit User Delete User |
| 21 | | Alex | Subts | client | Client | 2019-05-24 15:17:42 | 2019-05-24 15:17:59 | View User Edit User Delete User |
| 22 | | Нс | Нс | client | Client | 2019-05-24 18:58:50 | 2019-05-24 18:59:07 | View User Edit User Delete User |

10 user total

Alex's Information

[Back to Users](#)



Alex Smith
test@ag.com

Email: test@ag.com

First Name: Alex

Last Name: Smith

Last Name: Smith

City: Kyiv

Type: Client

Phone: 38063888522

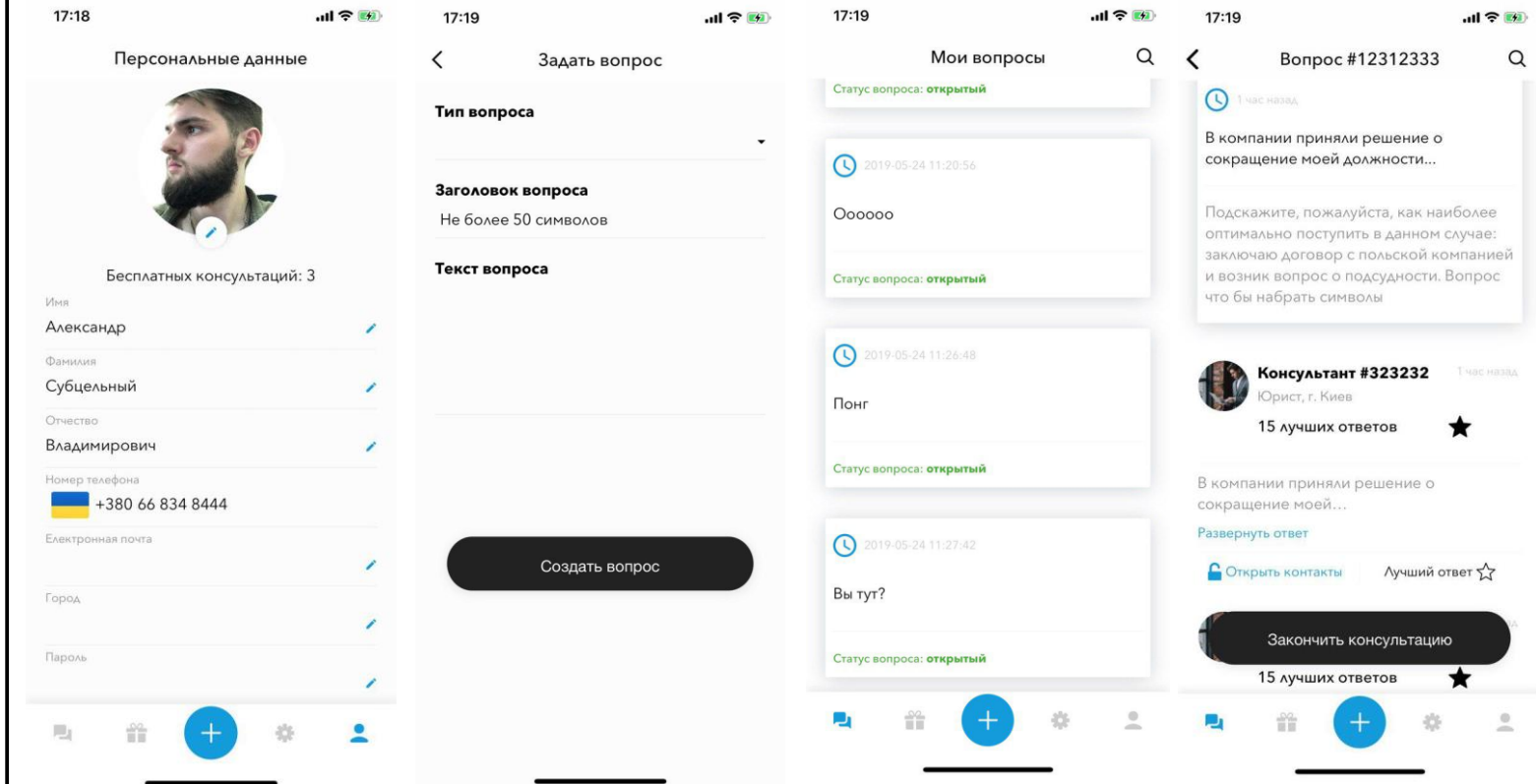
Role: User

Status: Active

Created At: 2019-05-24 15:30:20

Updated At: 2019-05-27 20:09:59

Email Signup IP: 31.134.127.144



| | | | | | | | | |
|-----------|------|-----------------|--------|------|---|--------|------|---------|
| | | | | | КП.ІП-5121.045440.06.99.КЕ | | | |
| | | | | | Креслення вигляду екранних форм | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| Зм. | Арк. | № документа | Підпис | Дата | | Літера | Маса | Масштаб |
| Розробив | | Субцельний О.В. | | | | | | |
| Перевірів | | Муха І.П. | | | | | | |
| Т. кон. | | | | | | Аркуш | | Аркушів |
| | | | | | Онлайн-платформа для надання юридичних послуг | | | |
| Н. кон. | | Ліщук К.І. | | | | | | |
| Затвердив | | Муха І.П. | | | | | | |
| | | | | | КП ім.Ігоря СікорськогоКафедра АСОІУпр. ІП-51 | | | |